Apache Kafka and KSQLDB in Action: Let's Build a Streaming Data Pipeline!

Dan Moff

Kafka Summit 2020

$ whoami

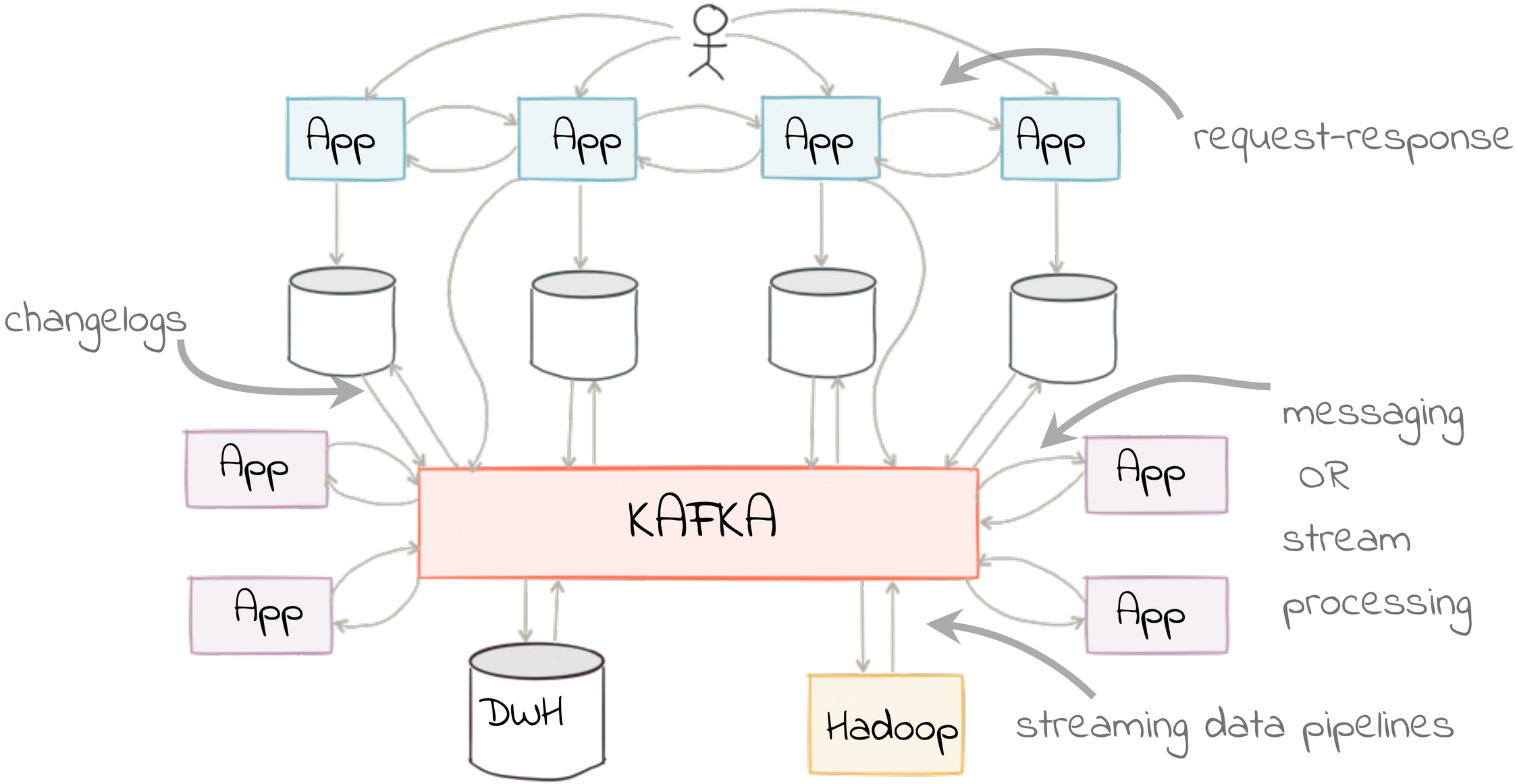
- Robin Moffatt (@rmoff)
- Senior Developer Advocate at Confluent

  (Apache Kafka, not Wikis 😉)

- Working in data & analytics since 2001
- Oracle ACE Director (Alumnus)

http://rmoff.dev/**talks** • http://rmoff.dev/**blog** • http://rmoff.dev/**youtube**

Kafka is an Event Streaming Platform



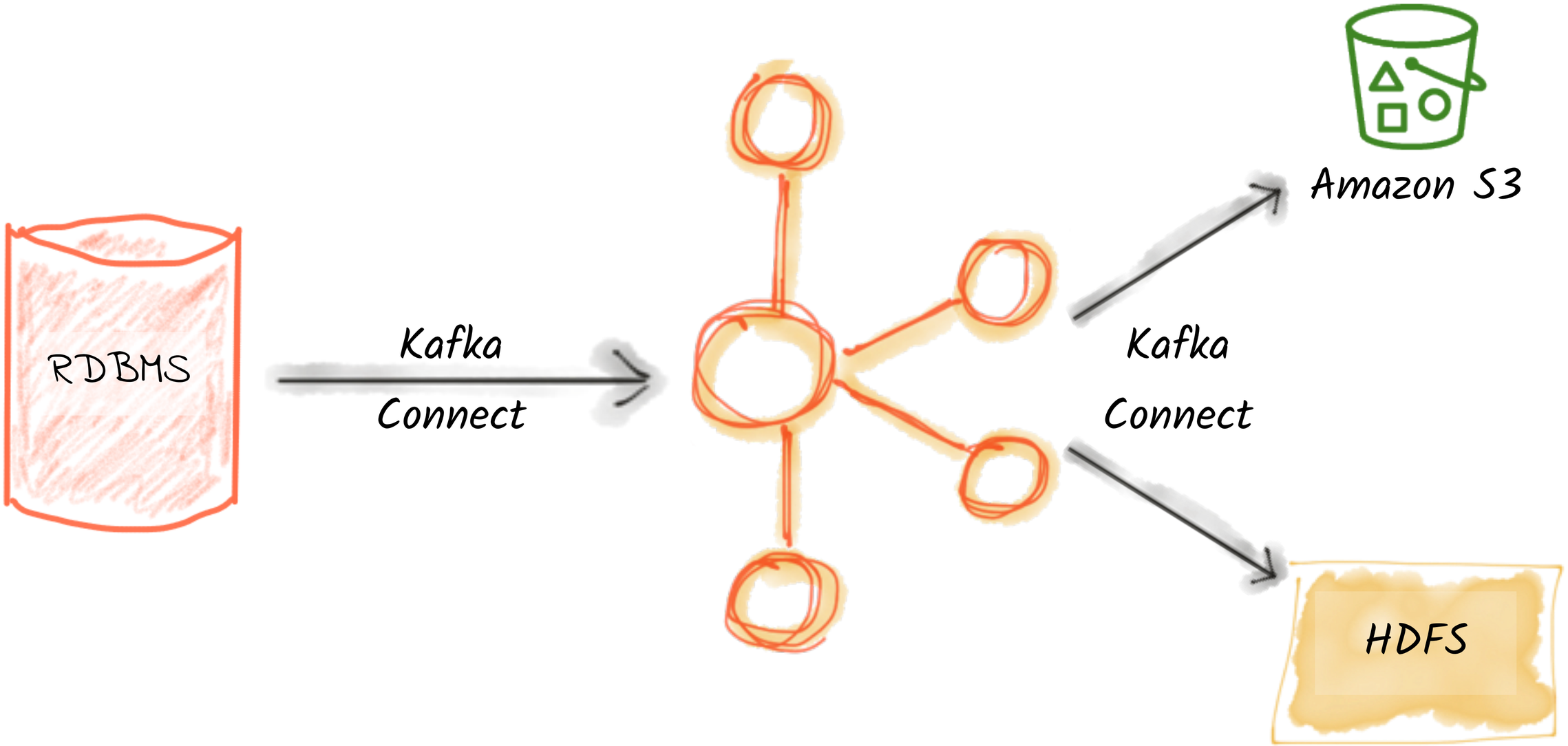
CONFLUENT

Streaming Data Pipelines

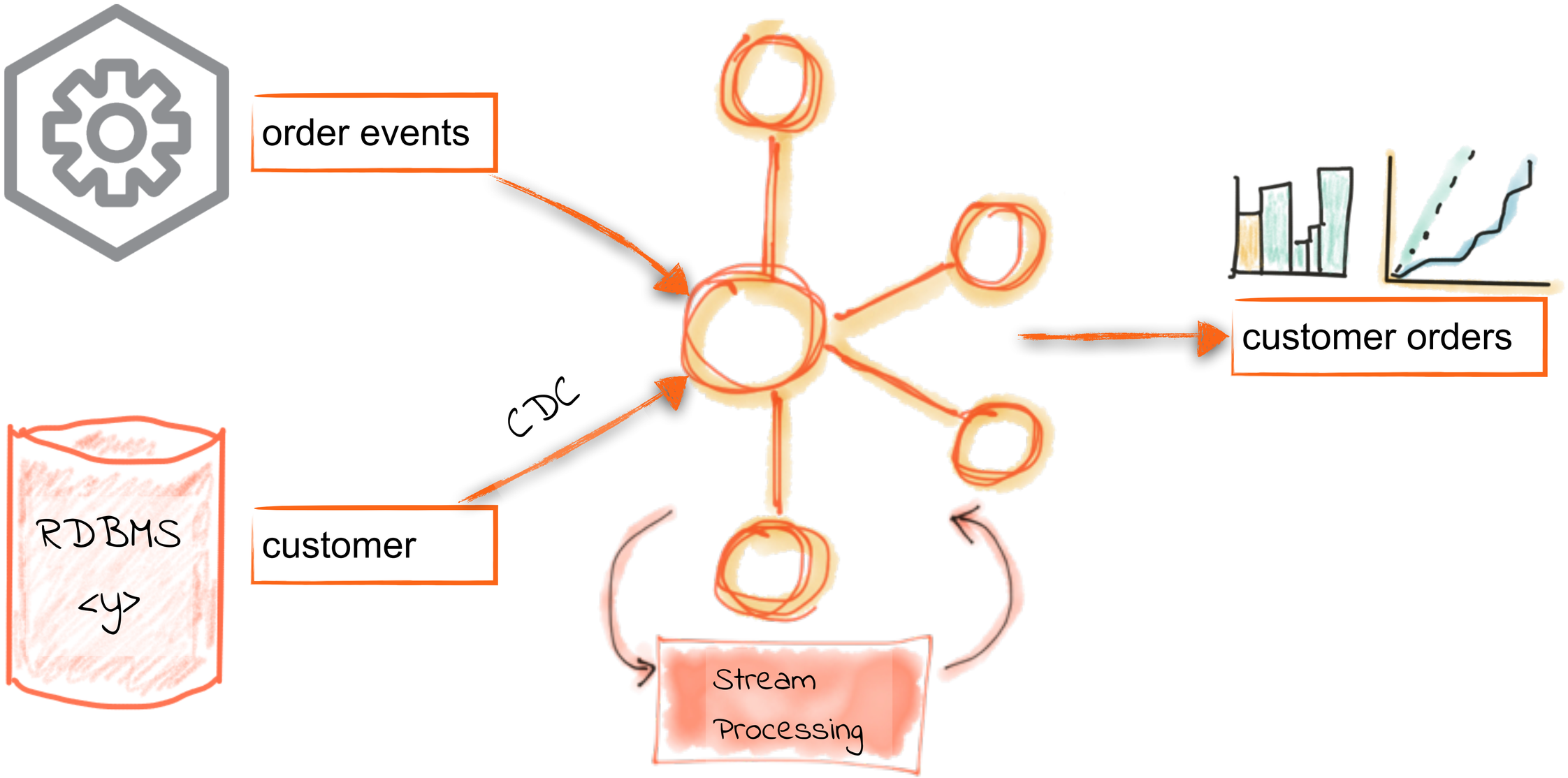
@rmoff | #ConfluentVUG | @confluentinc

Photo by Victor Garcia on Unsplash

Database Offload



Real-time Event Stream Enrichment



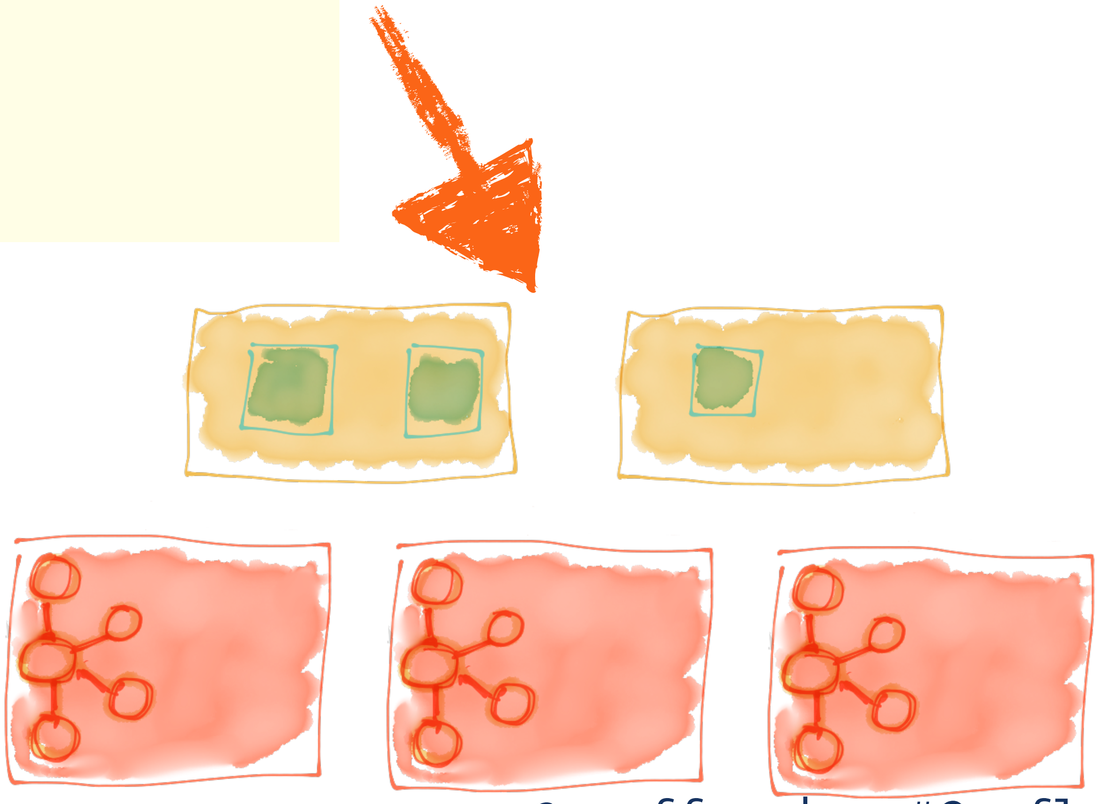
Building a streaming data pipeline

Integration + Stream Processing

Integration

Streaming Integration with Kafka Connect

Sources

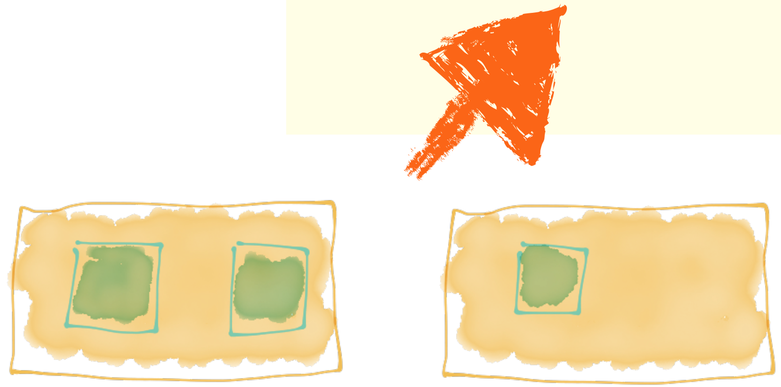


Kafka Connect

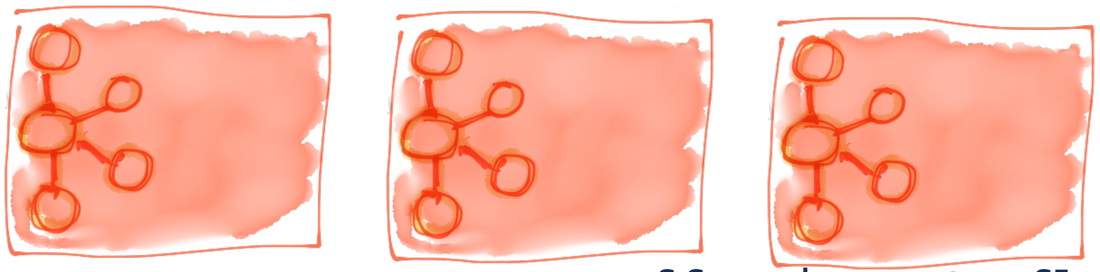
Kafka Brokers

Streaming Integration with Kafka Connect

Sinks

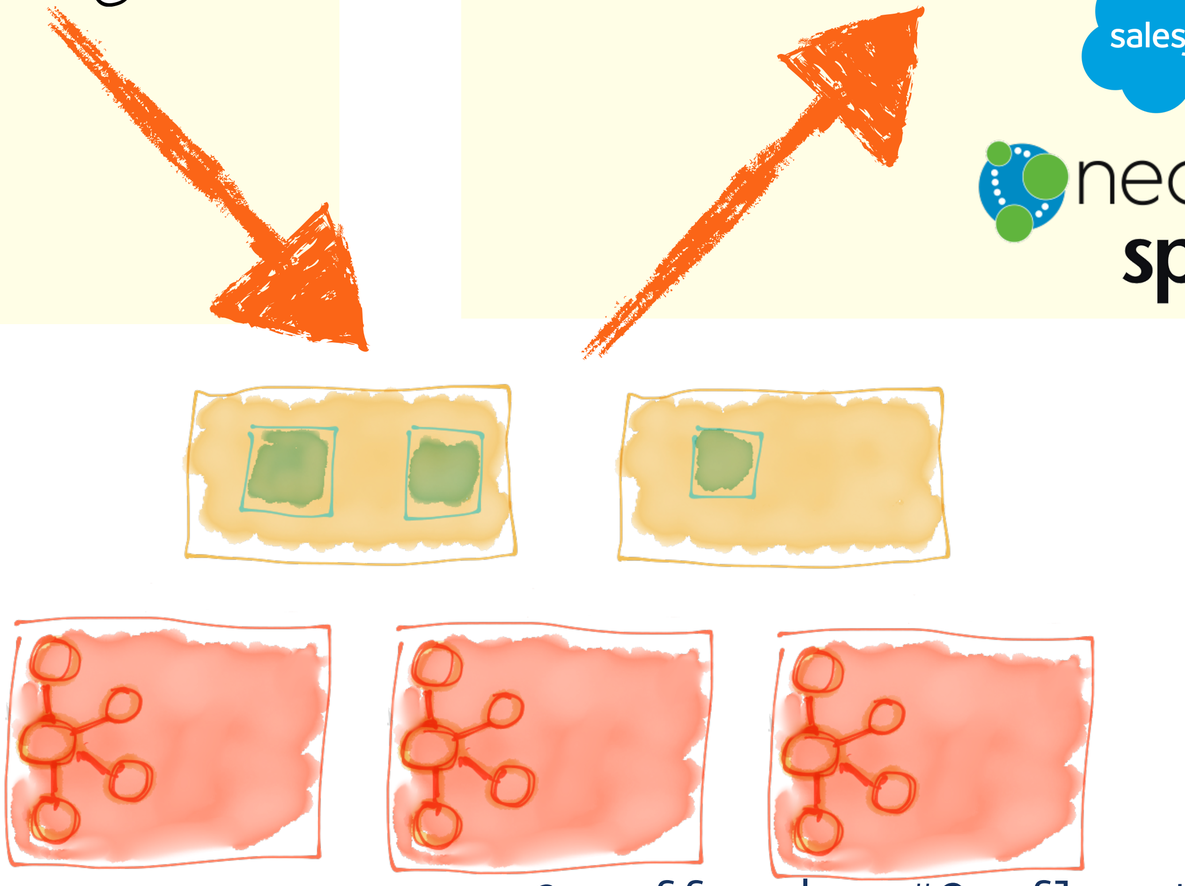


Kafka Connect



Kafka Brokers

Streaming Integration with Kafka Connect



Kafka Connect

Kafka Brokers

Look Ma, No Code!

```
{
  "connector.class":
    "io.confluent.connect.jdbc.JdbcSourceConnector",
  "connection.url":
    "jdbc:mysql://asgard:3306/demo",
  "table.whitelist":
    "sales,orders,customers"
}
```

Serialisation & Schemas

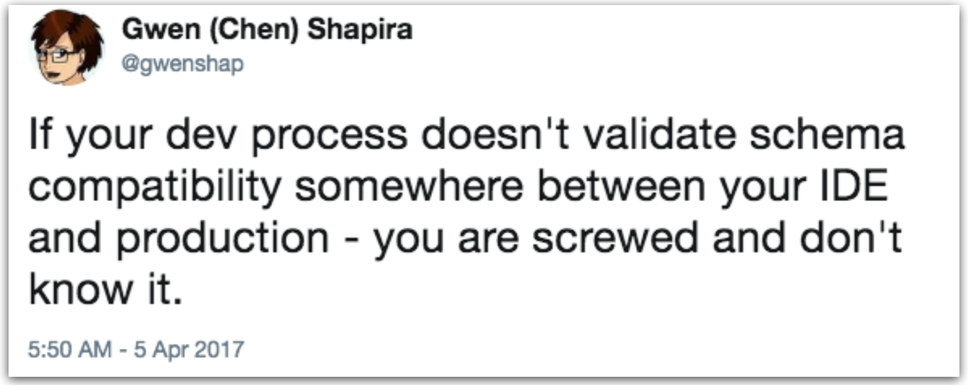
Avro 👍

Protobuf 👍

JSON Schema 👍

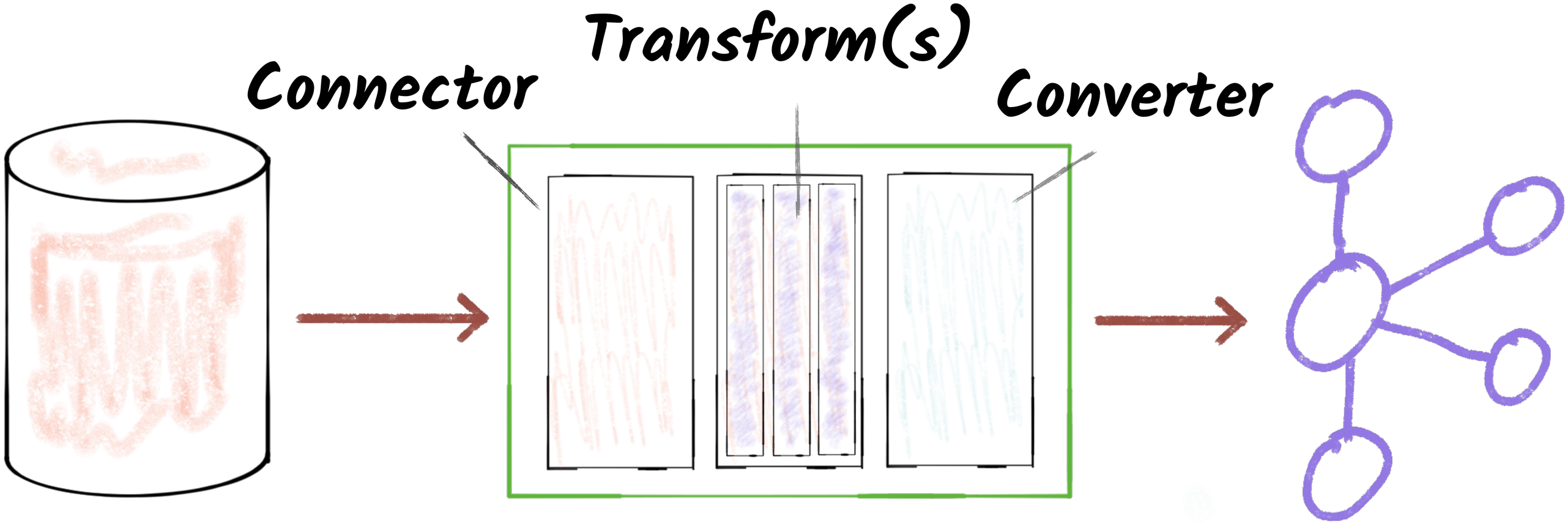
JSON 😬

CSV

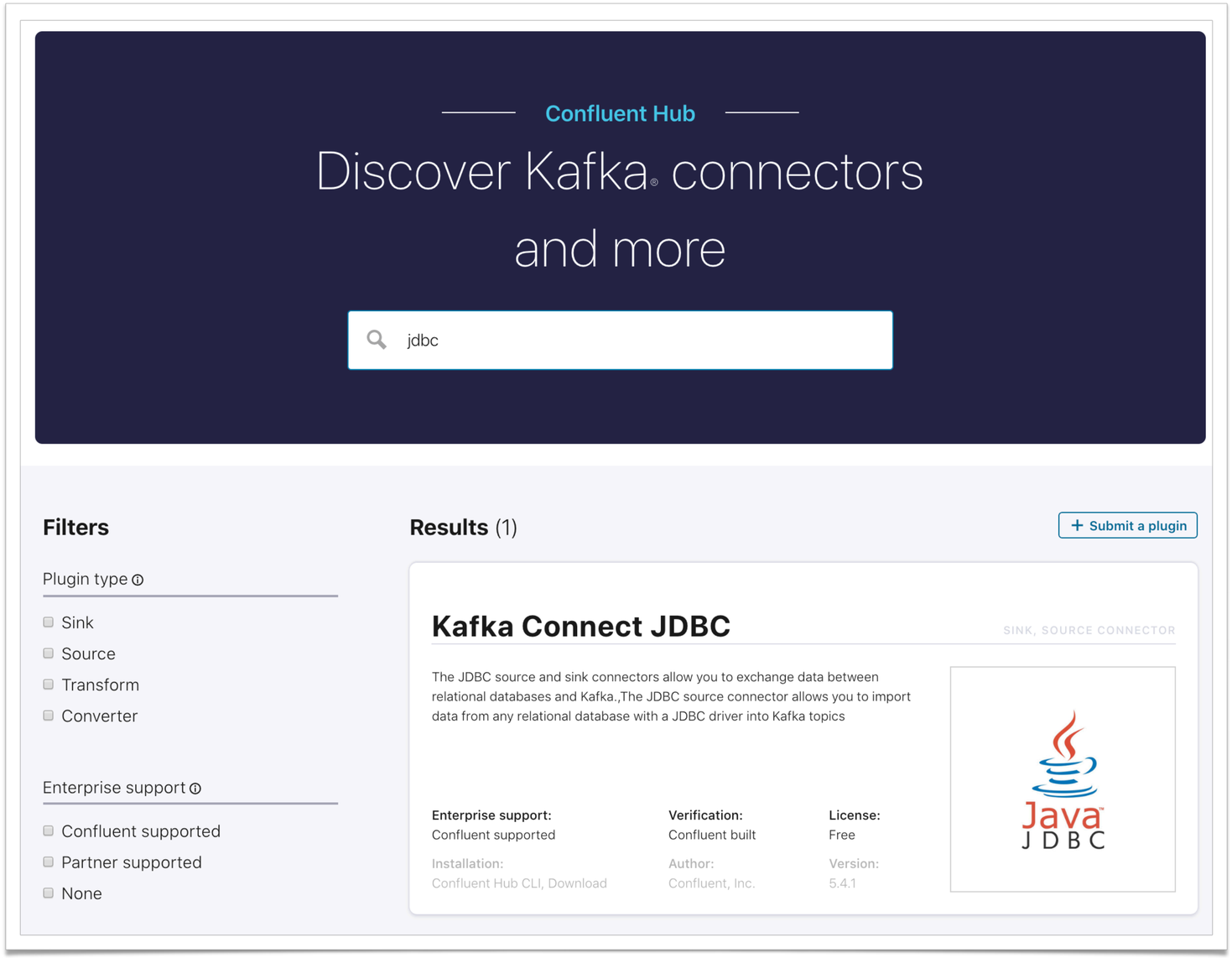


https://rmoff.dev/qcon-schemas

Extensible

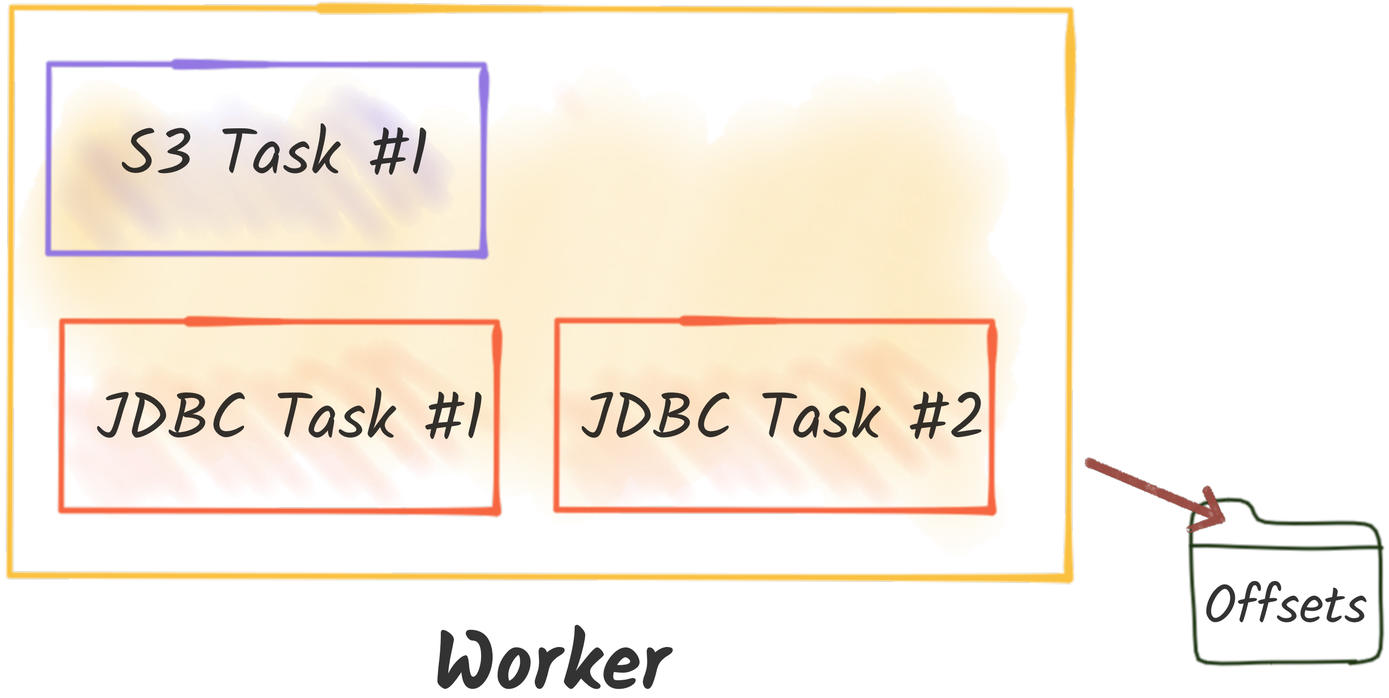


Confluent Hub

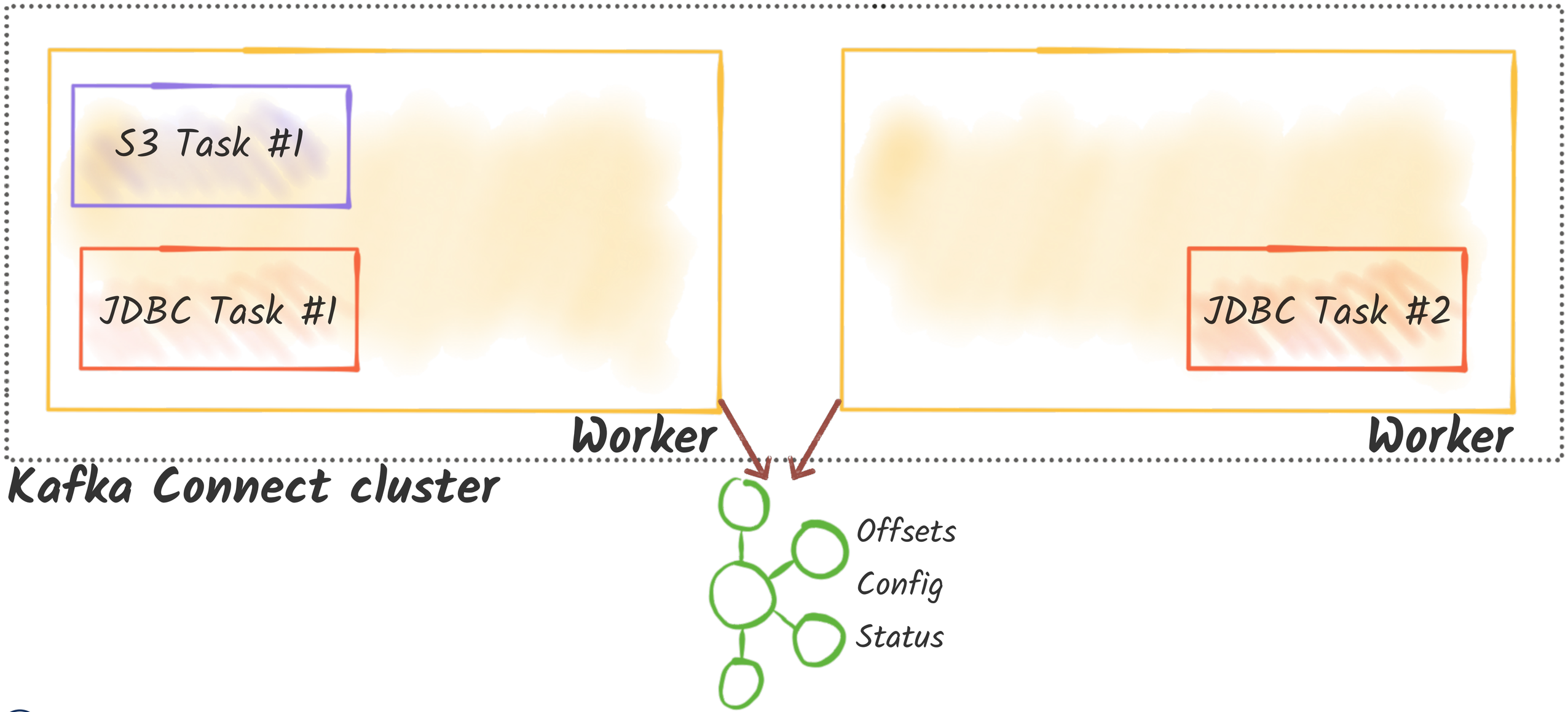


hub.confluent.io

Kafka Connect Standalone Worker

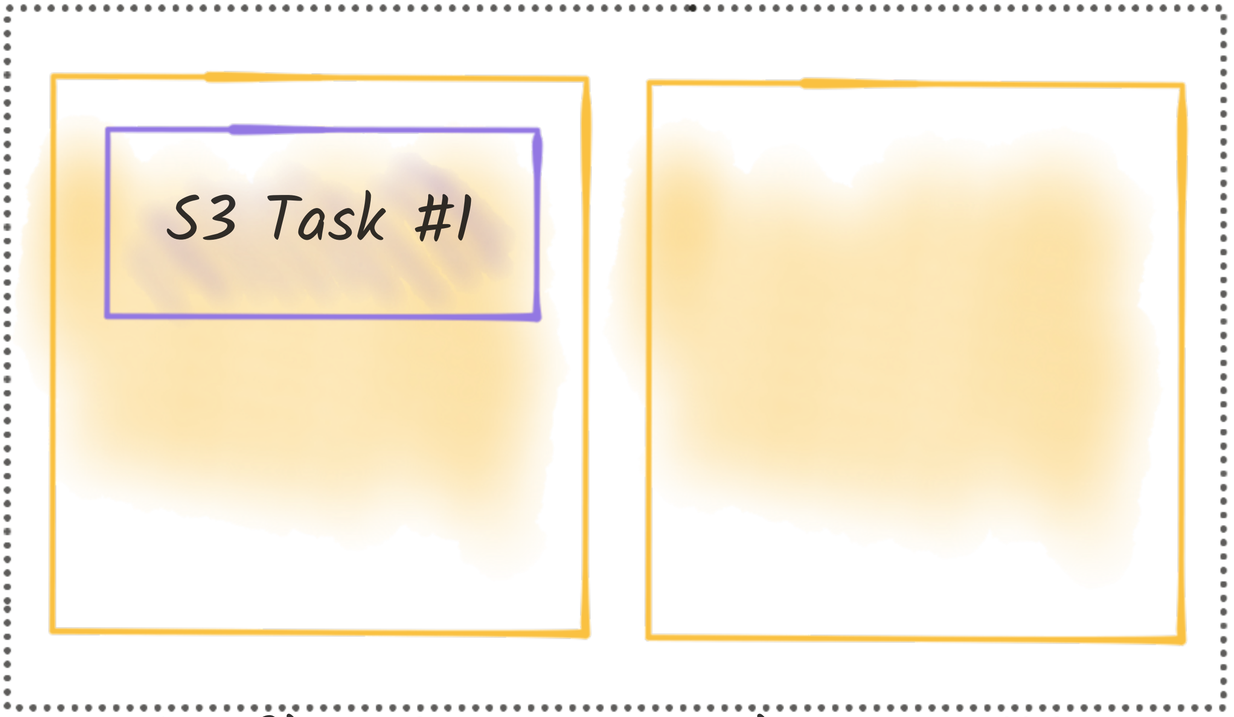


Kafka Connect Distributed Workers



Fault-tolerant? Yeah!

Multiple Distributed Clusters



Kafka Connect cluster #1

Kafka Connect cluster #2

Integration + Stream Processing

Stream Processing

Photo by Franck V. on Unsplash

```
SELECT *
FROM WIDGETS
WHERE WEIGHT_G > 120
```

```
SELECT COUNT(*)
FROM WIDGETS
GROUP BY PRODUCTION_LINE
```

```
SELECT AVG(TEMP_CELCIUS) AS TEMP
FROM WIDGETS
GROUP BY SENSOR_ID
HAVING TEMP>20
```

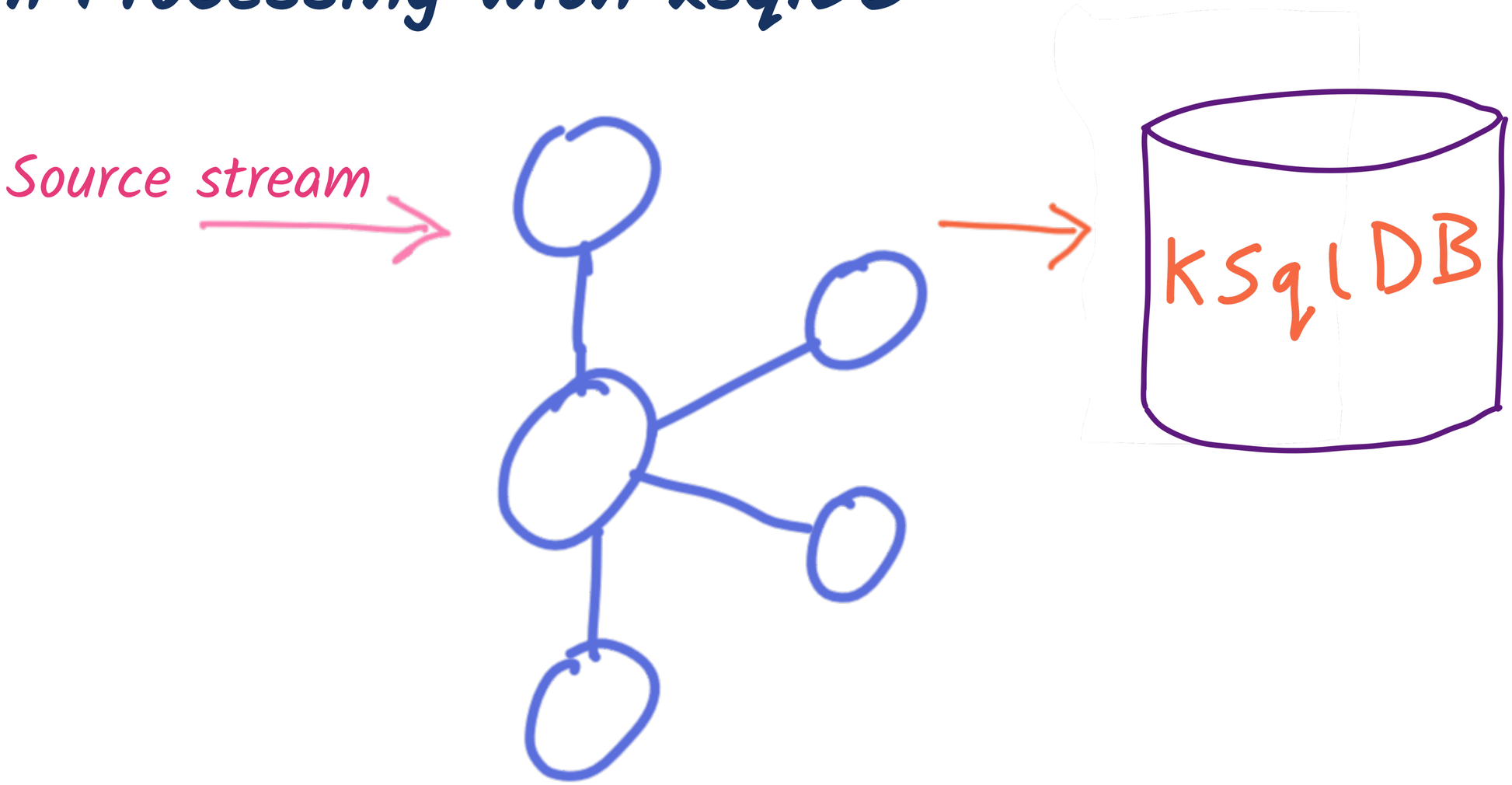
```
{
    "reading_ts": "2020-02-14T12:19:27Z",
    "sensor_id": "aa-101",
    "production_line": "w01",
    "widget_type": "acme94",
    "temp_celcius": 23,
    "widget_weight_g": 100
}
```

Object store, data warehouse, RDBMS

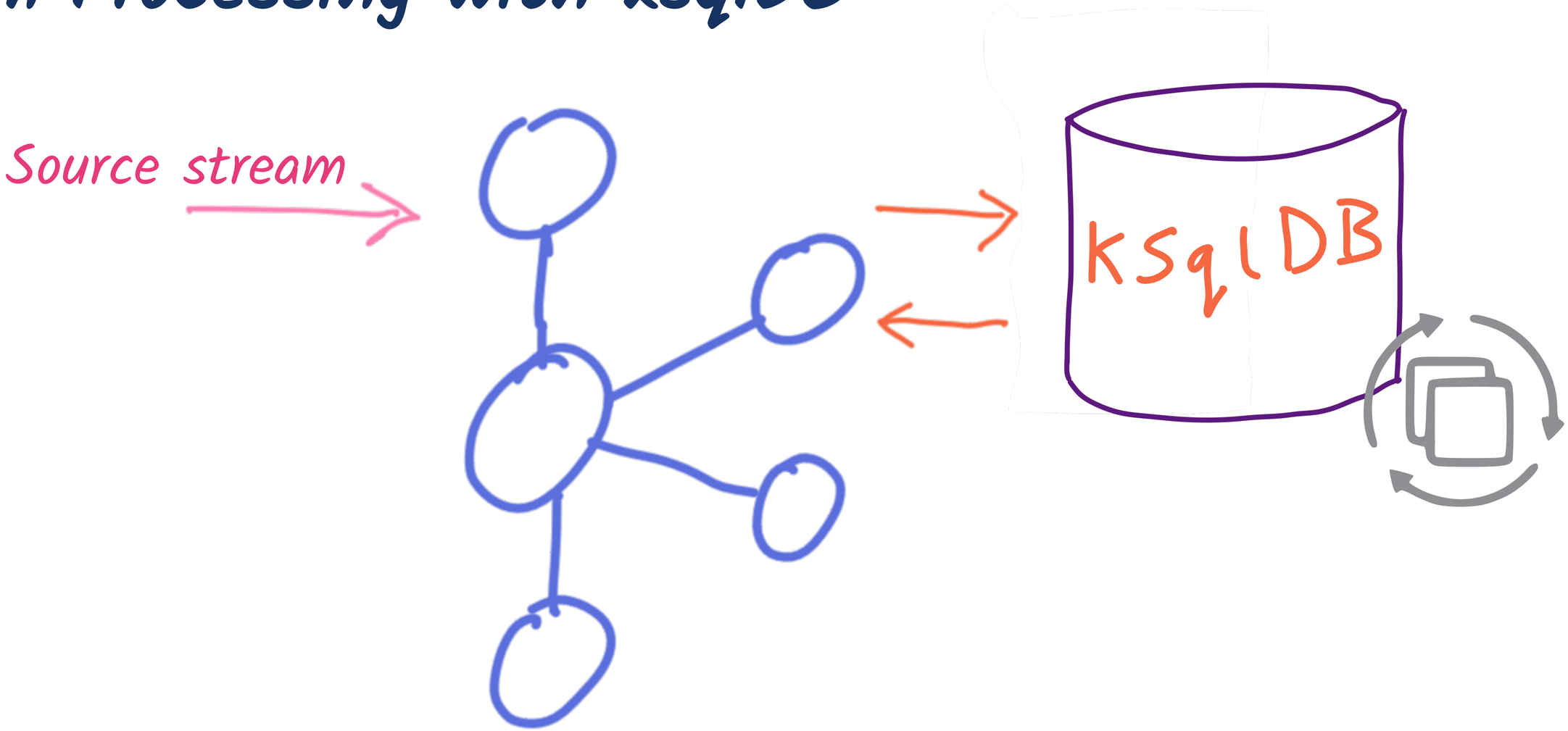
```
CREATE SINK CONNECTOR dw WITH (
'connector.class' = 'S3Connector',
'topics' = 'widgets'
…);
```

Photo by Franck V. on Unsplash

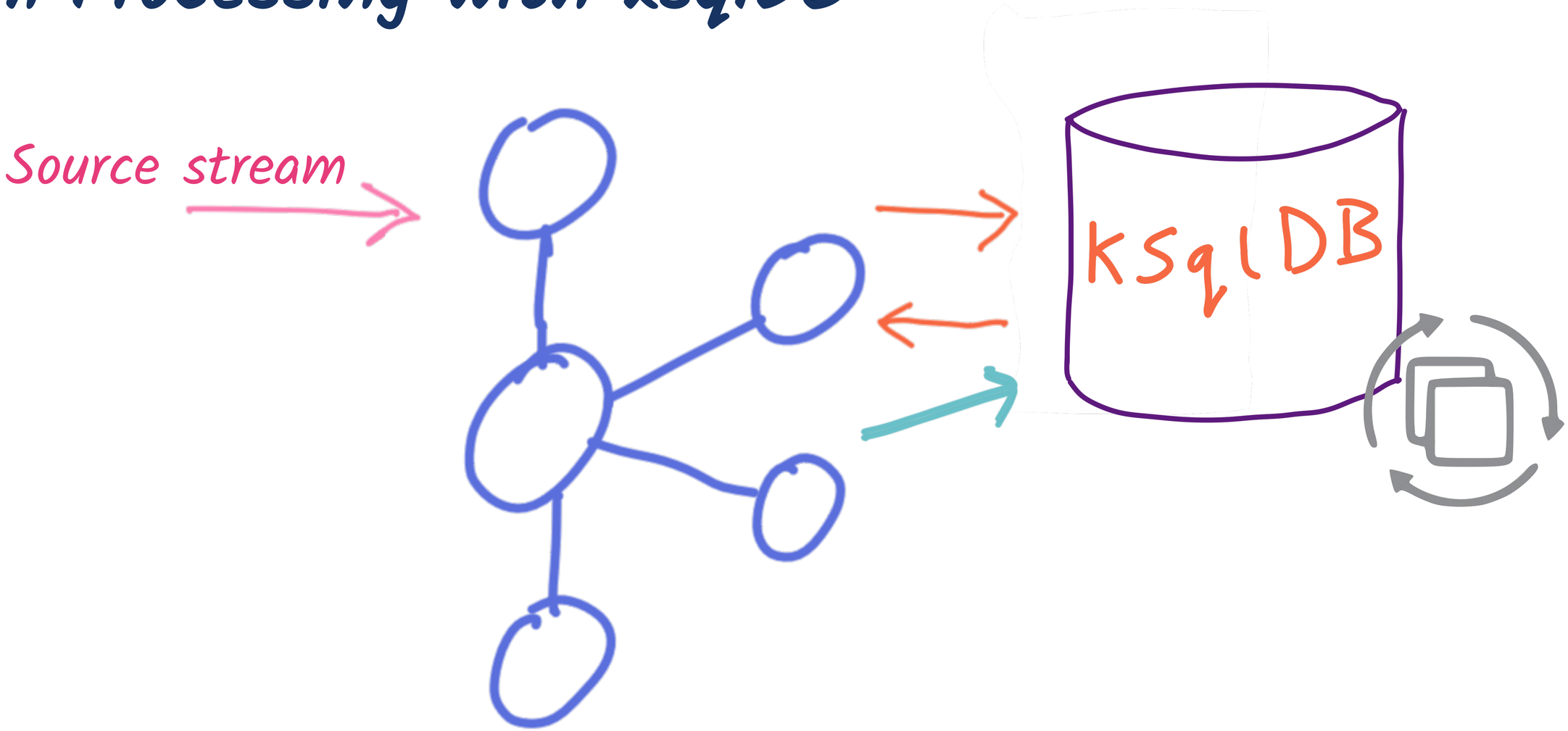
Stream Processing with ksqlDB



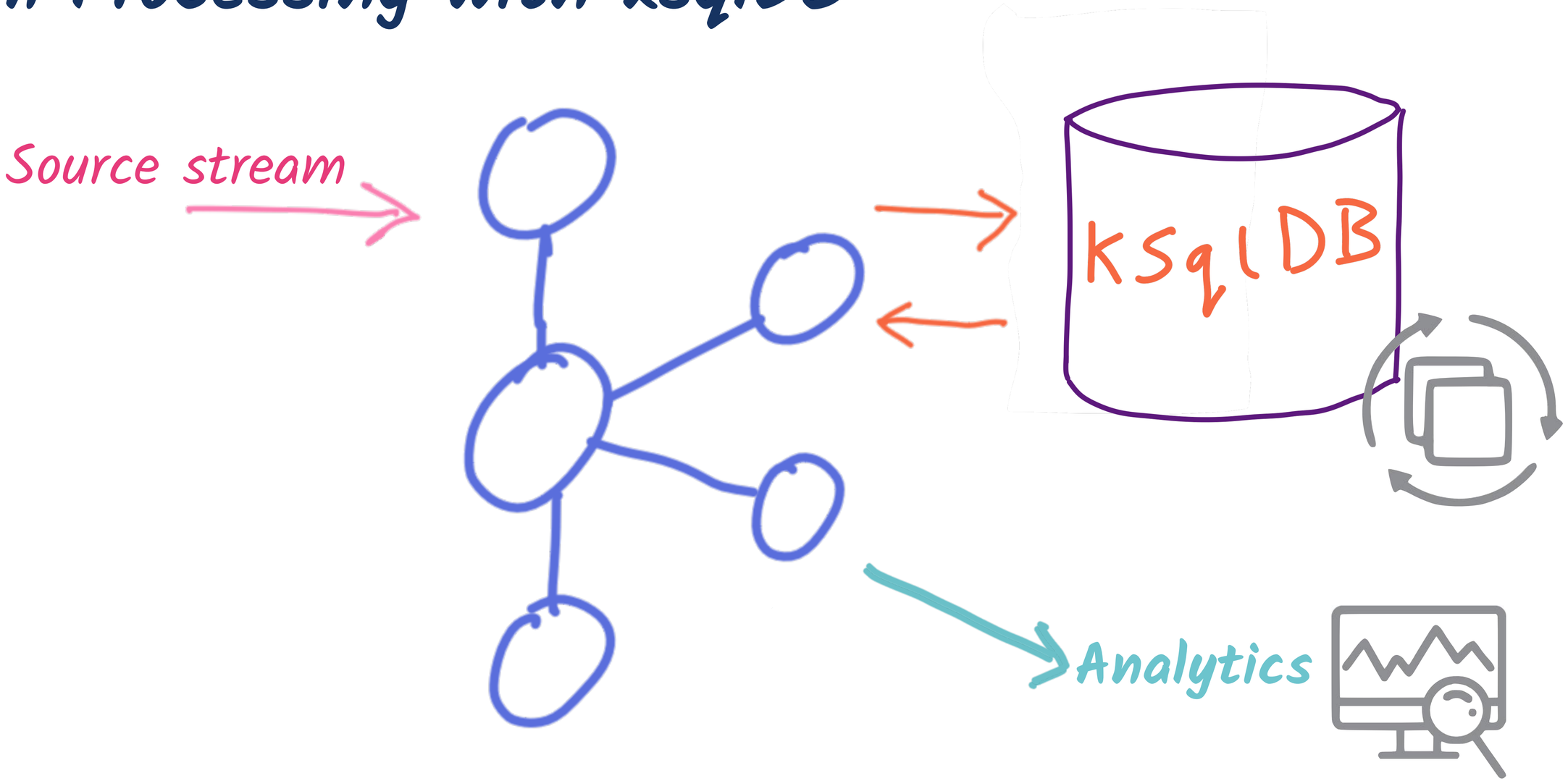
Stream Processing with ksqlDB



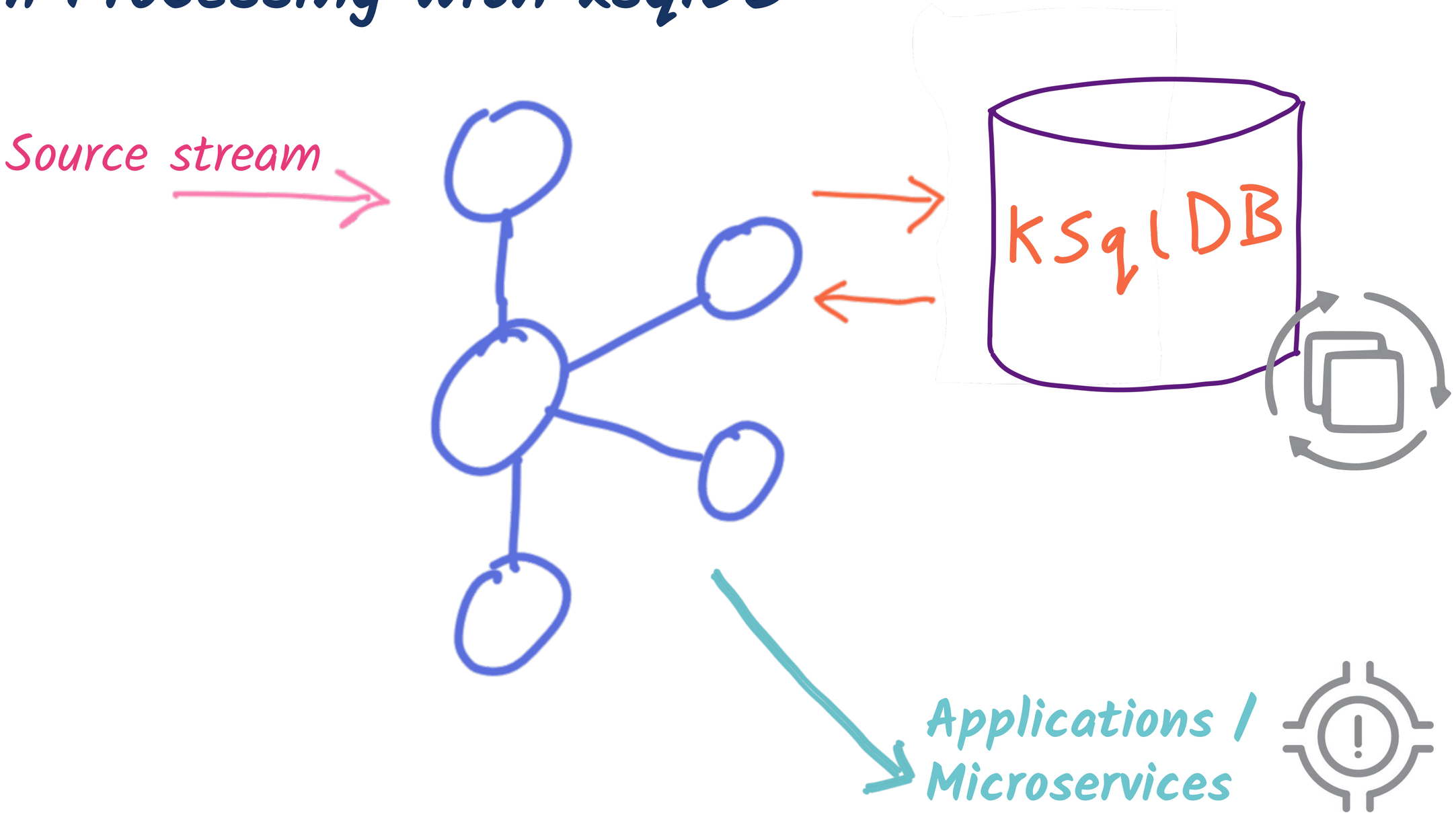
Stream Processing with ksqlDB



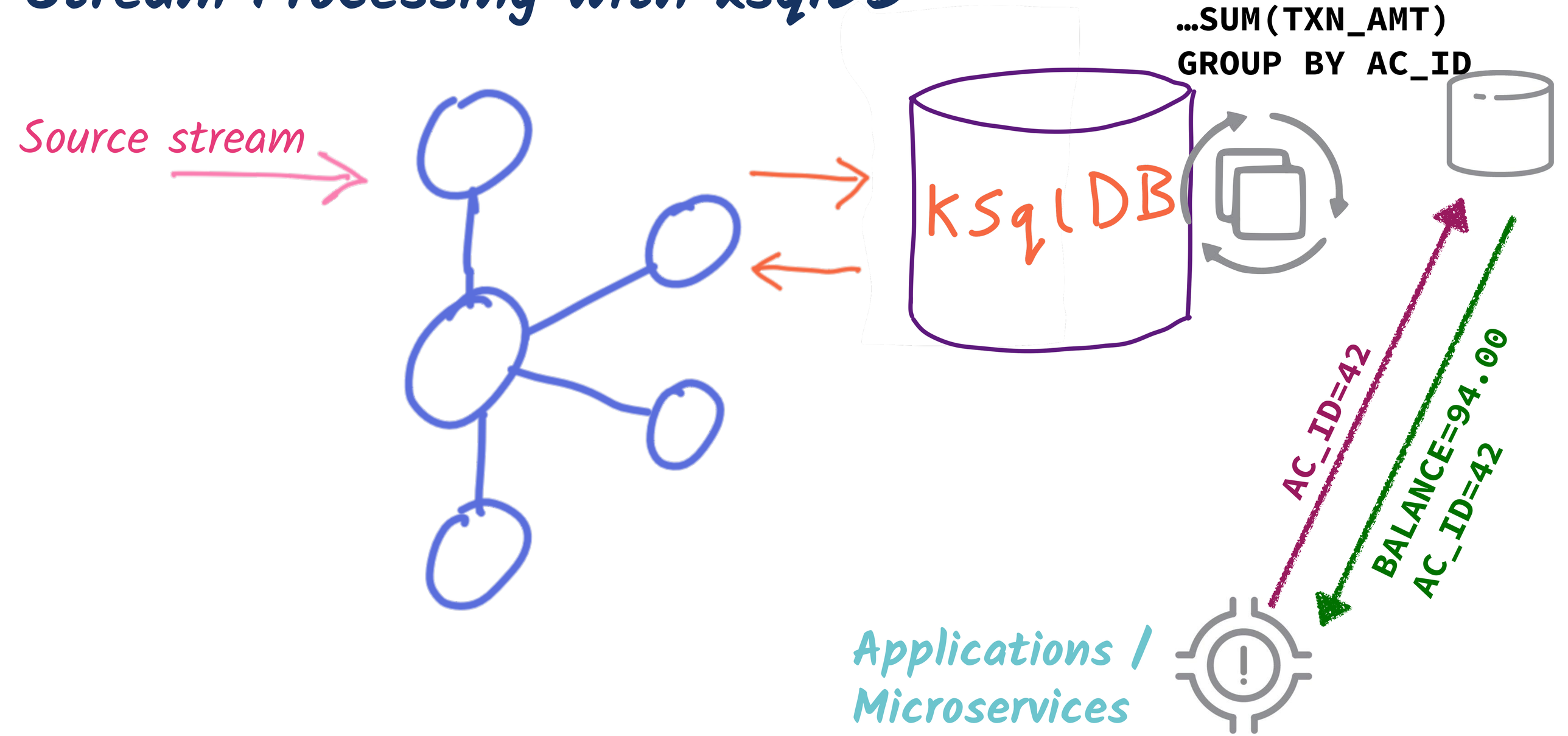
Stream Processing with ksqlDB



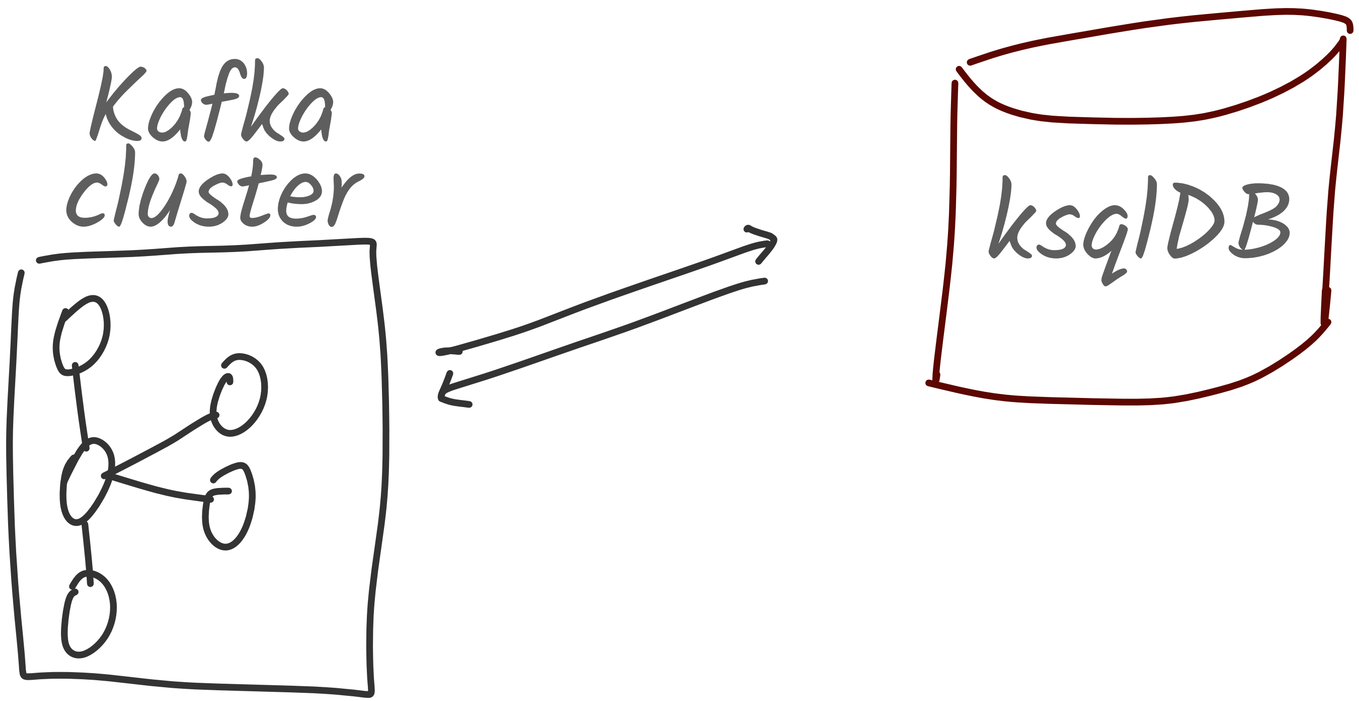
Stream Processing with ksqlDB



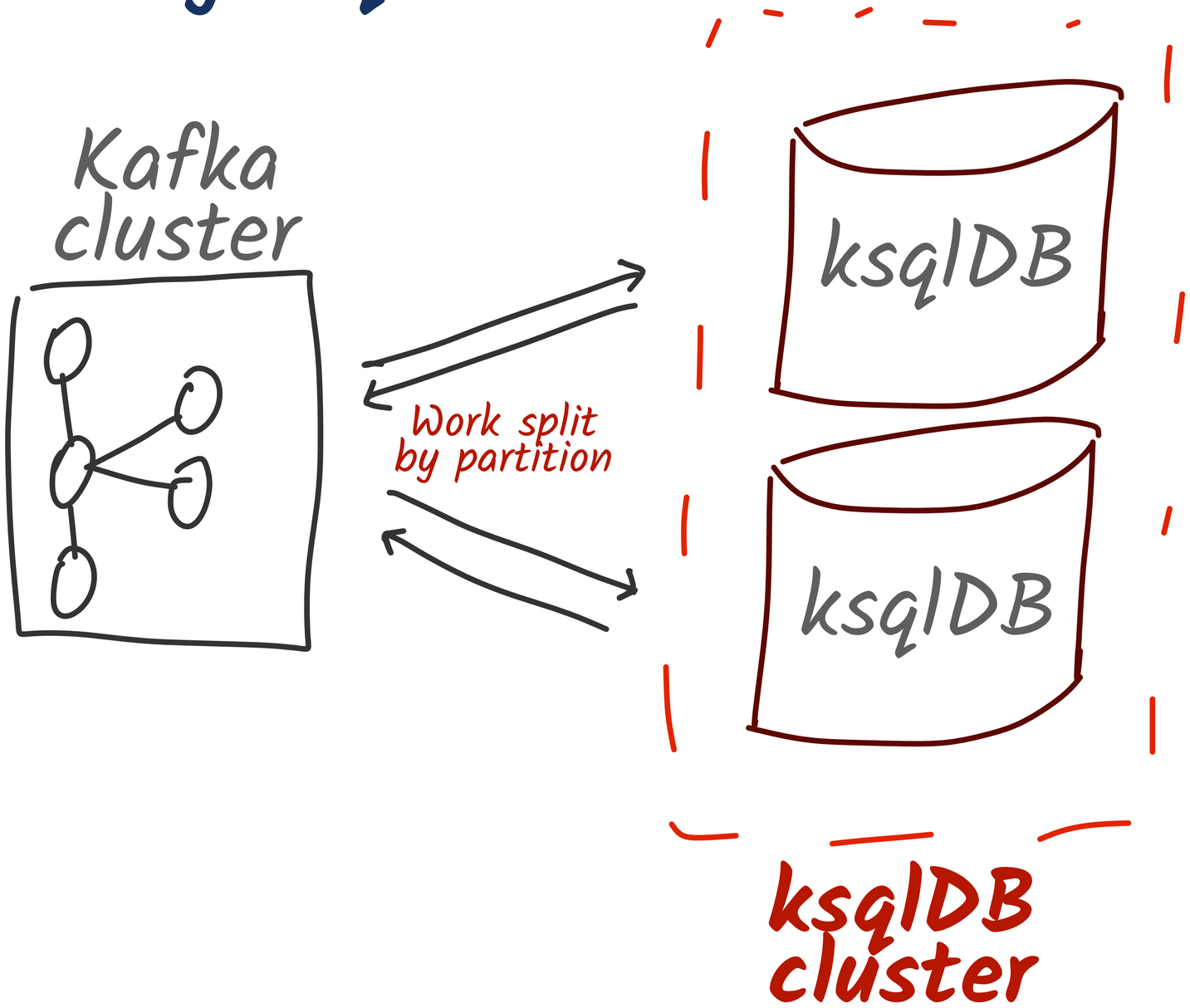
Stream Processing with ksqlDB



Scaling ksqlDB

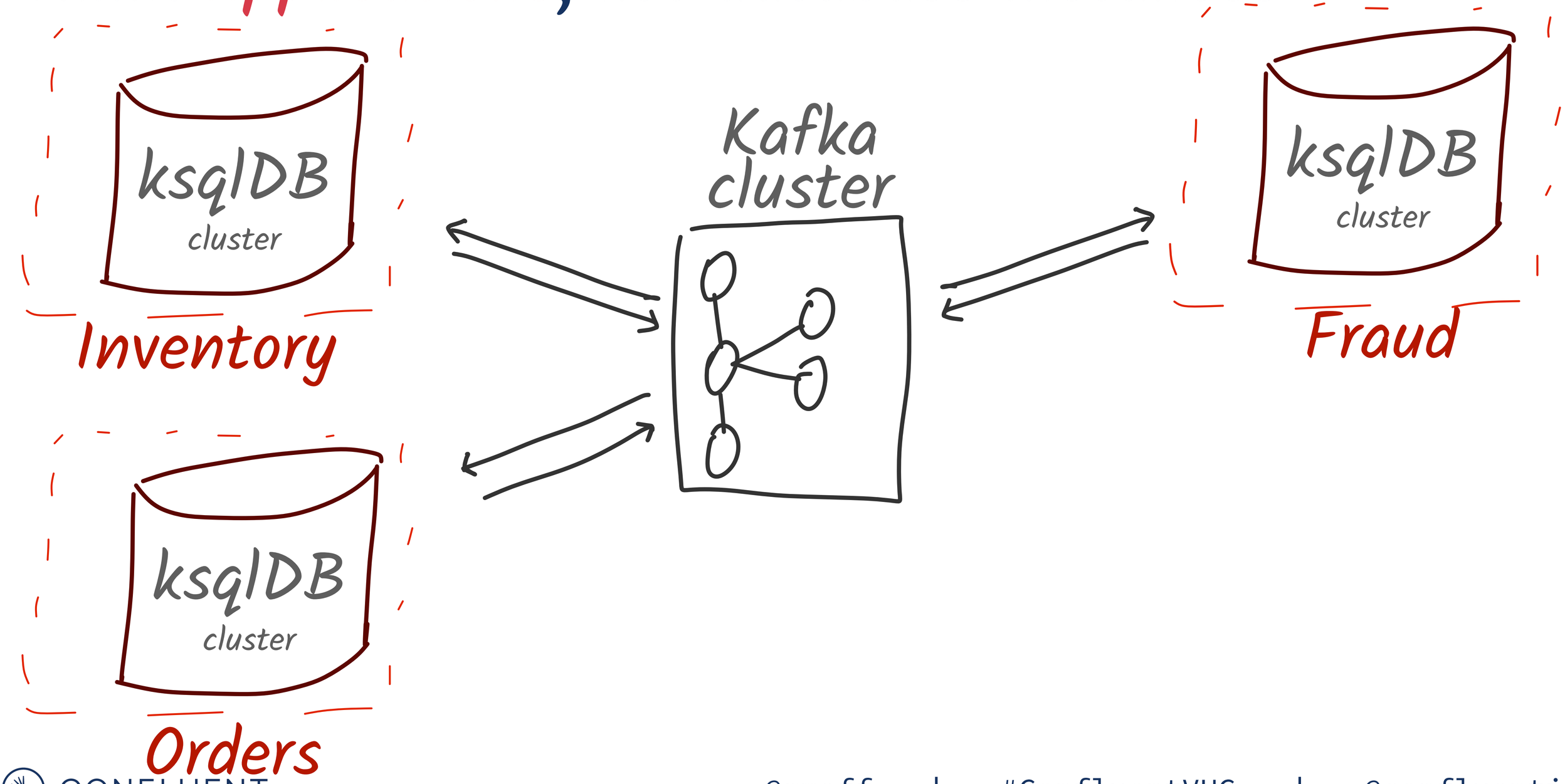


Scaling ksqlDB

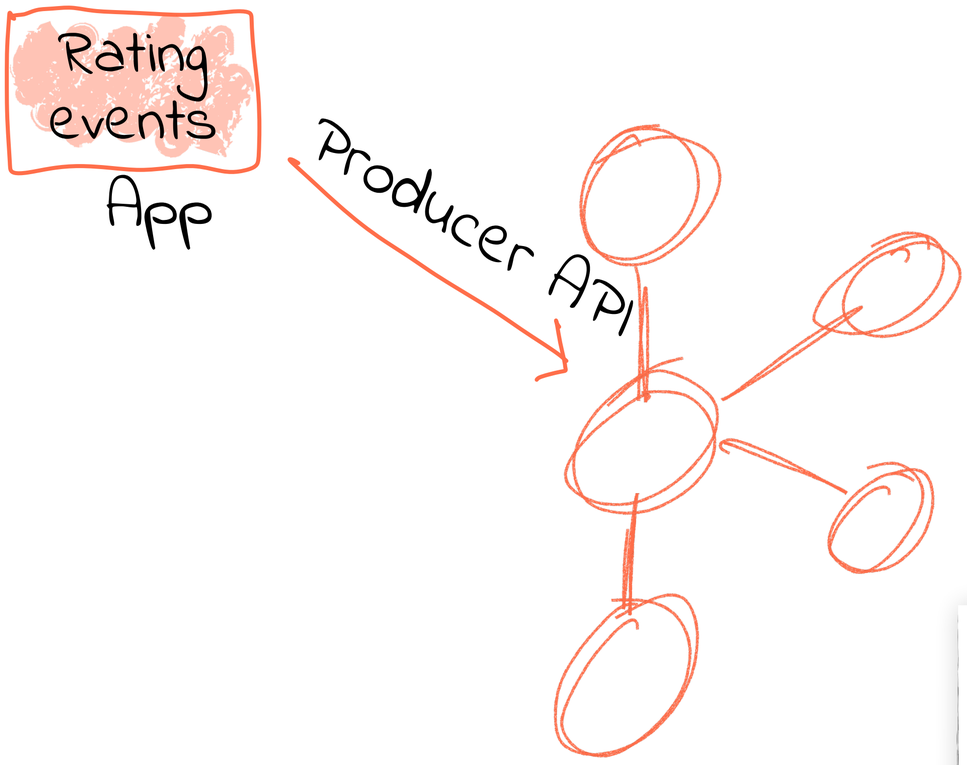


ONE DOES NOT SIMPLY RUN
ONE SINGLE KSQLDB CLUSTER

Think Applications, not database instances

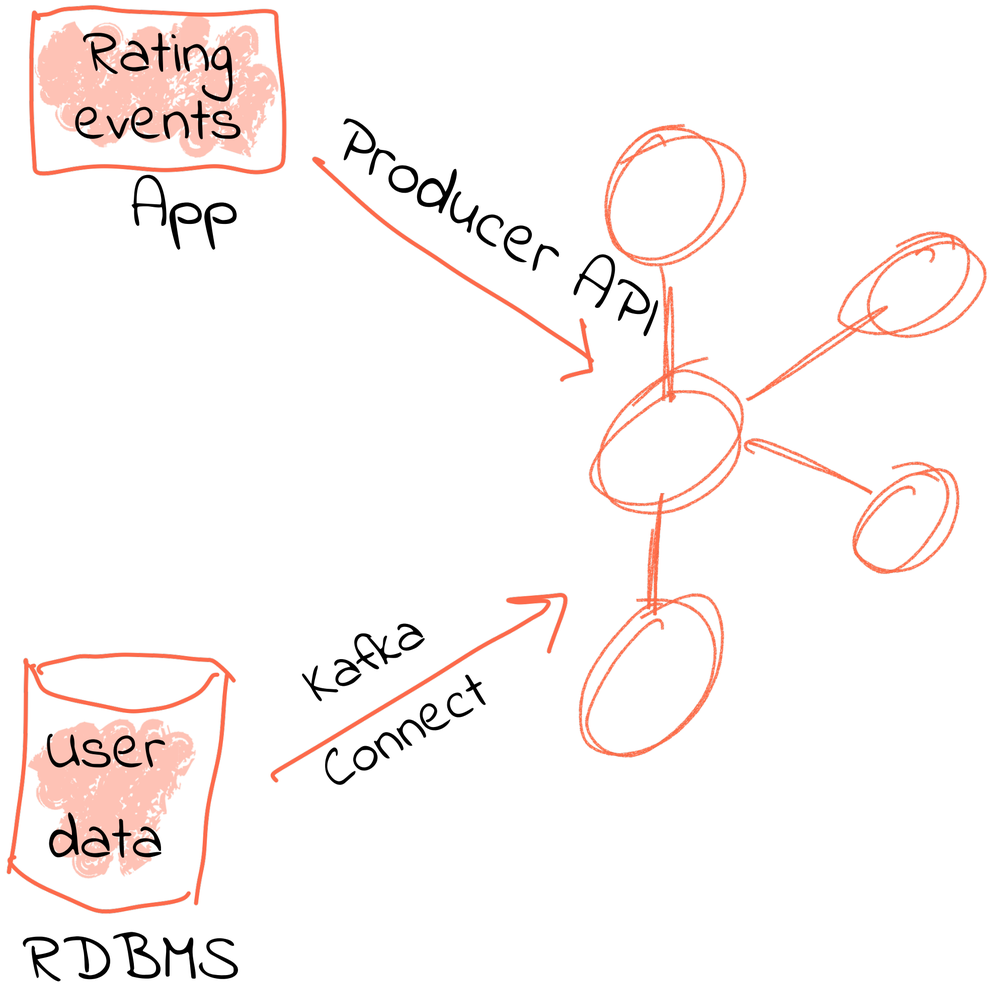


Let's Build It!

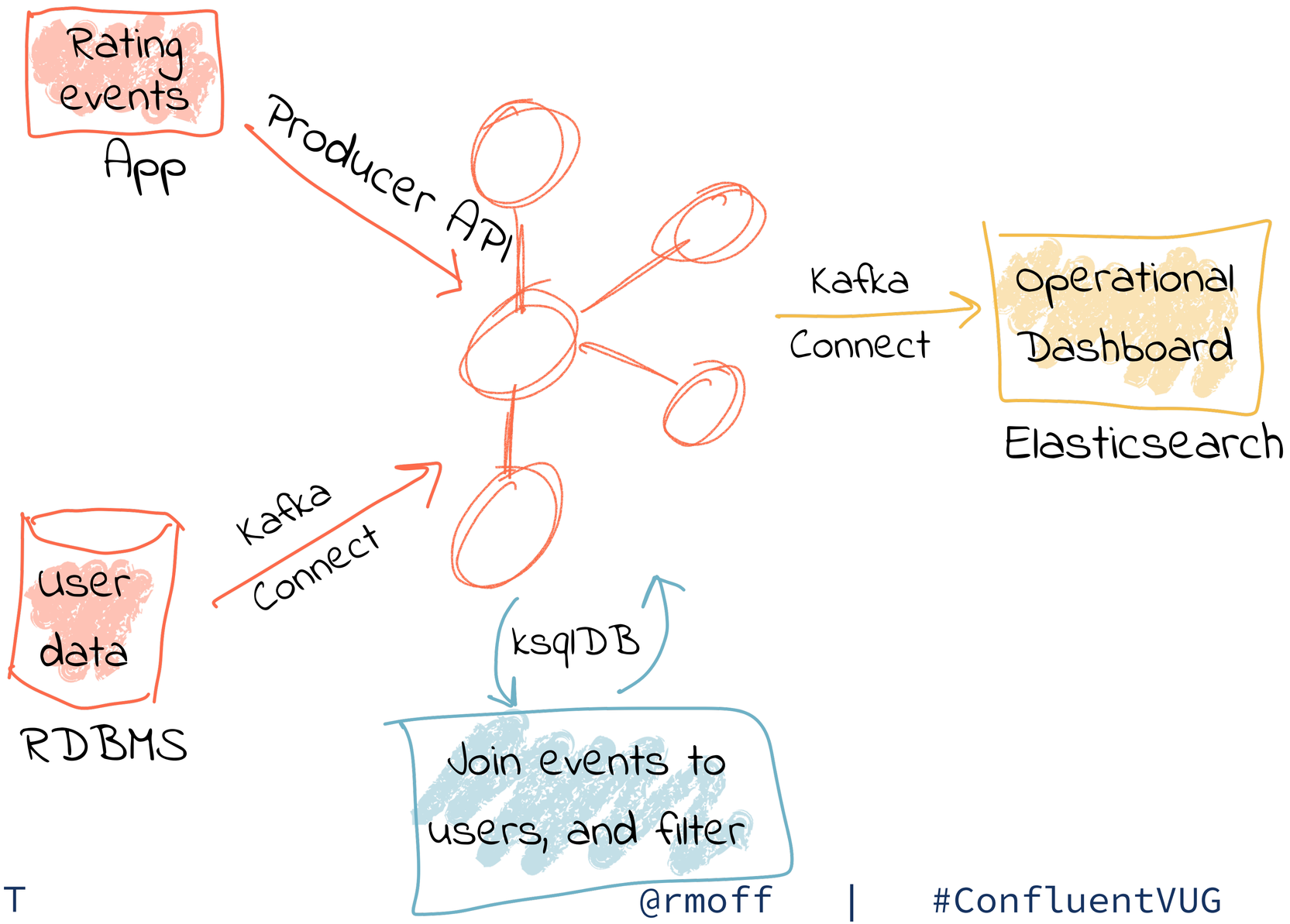



```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105213,
 "channel": "web",
 "message": "worst. flight. ever. #neveragain"
}
```

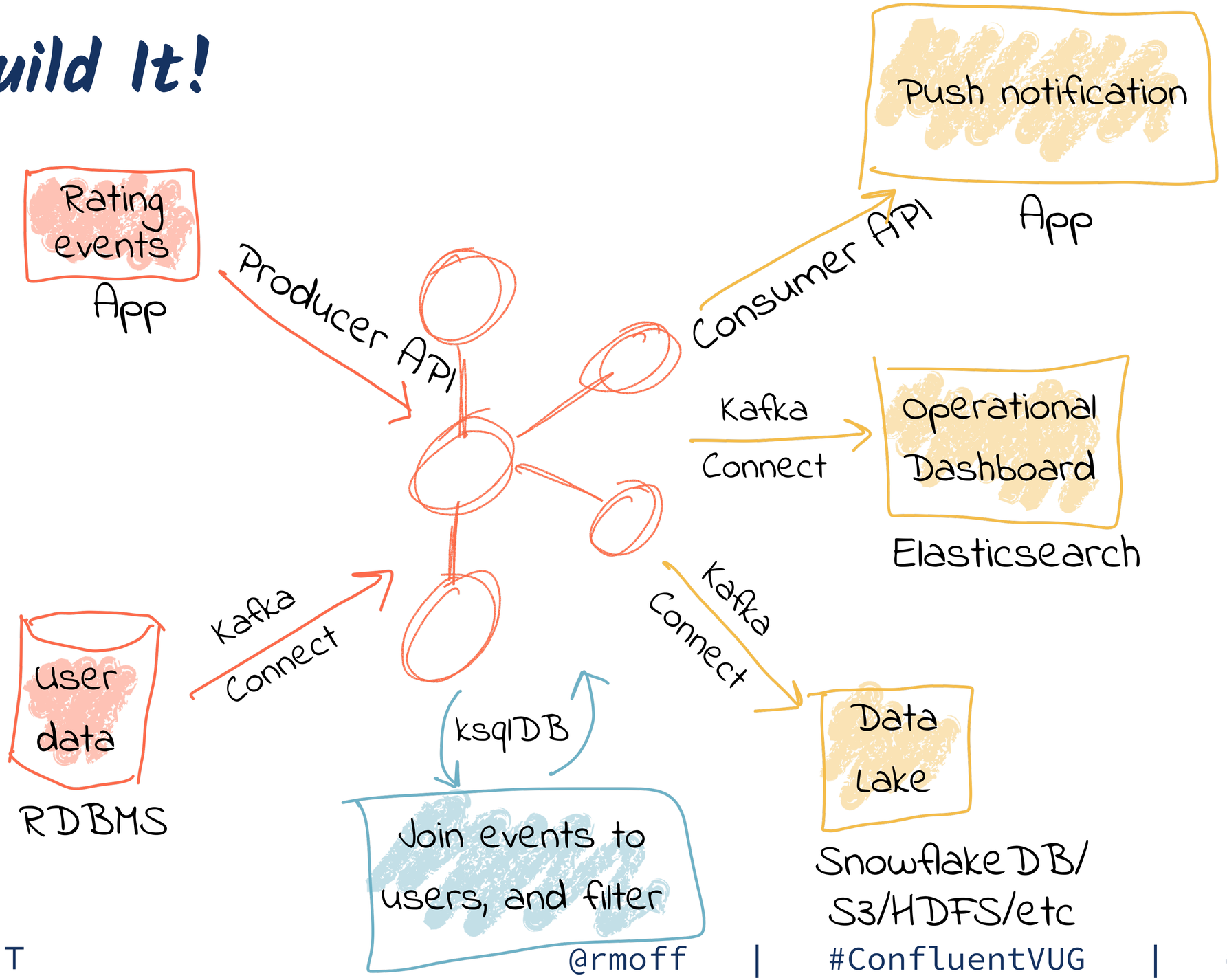
Let's Build It!



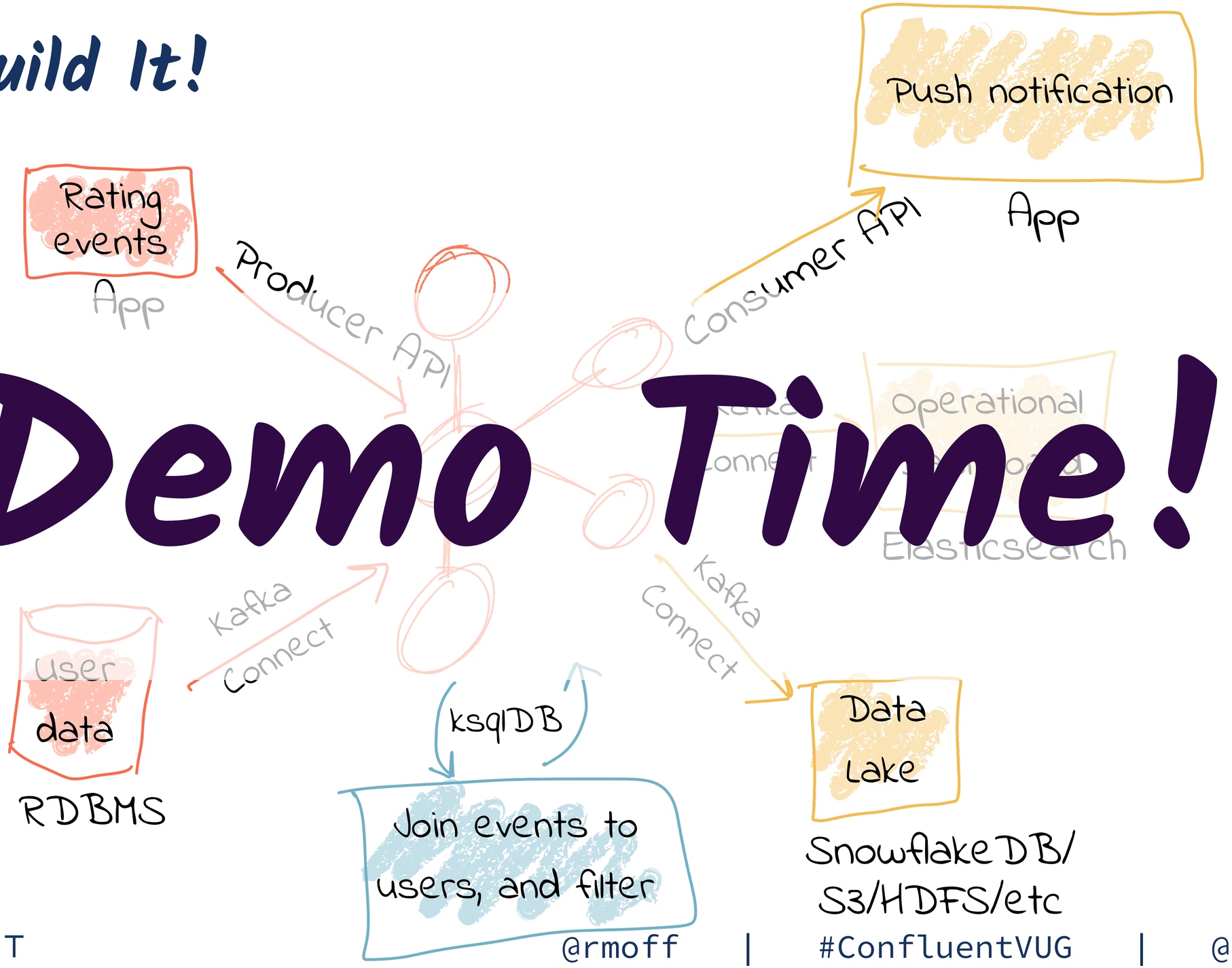
Let's Build It!

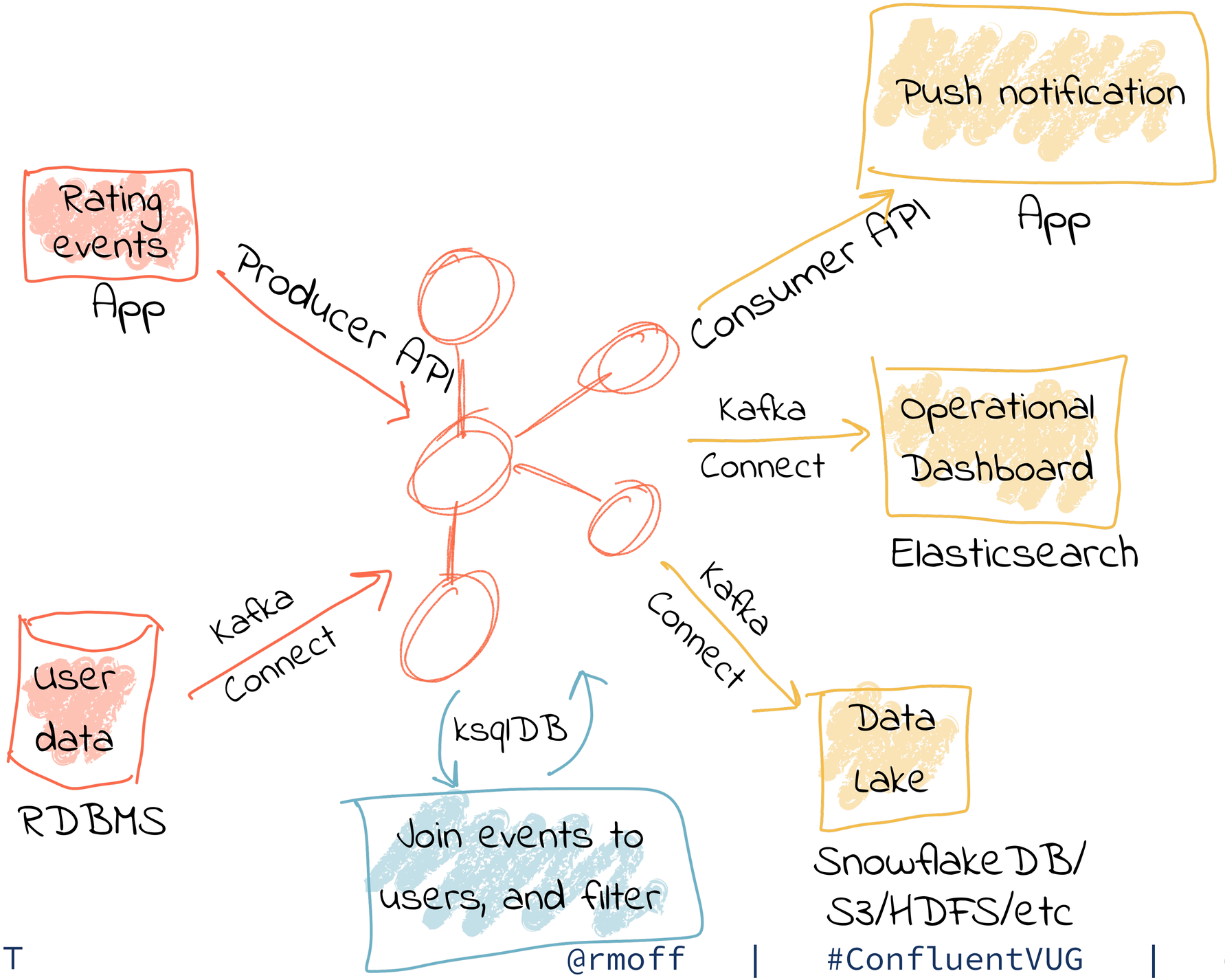


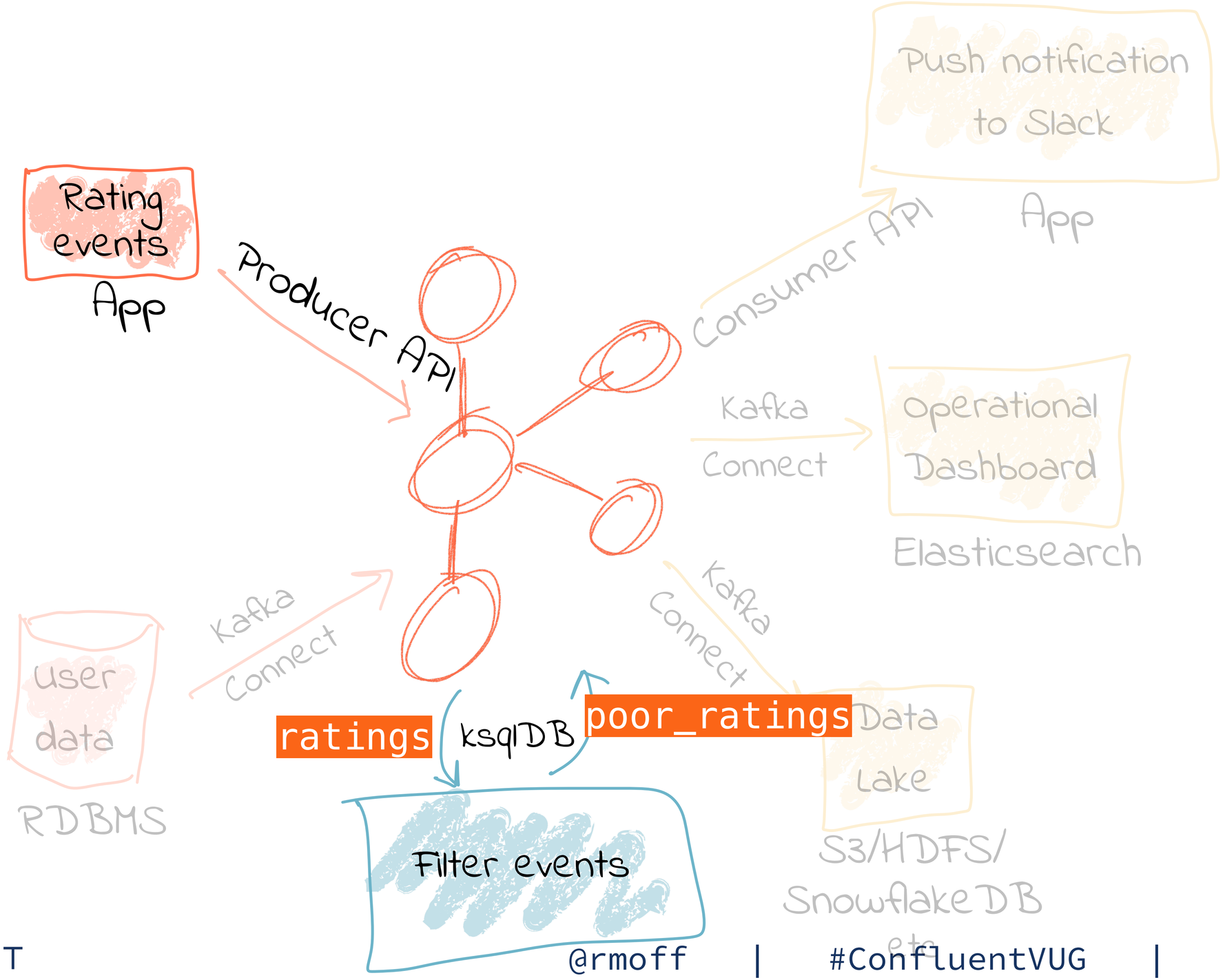
Let's Build It!



Let's Build It!

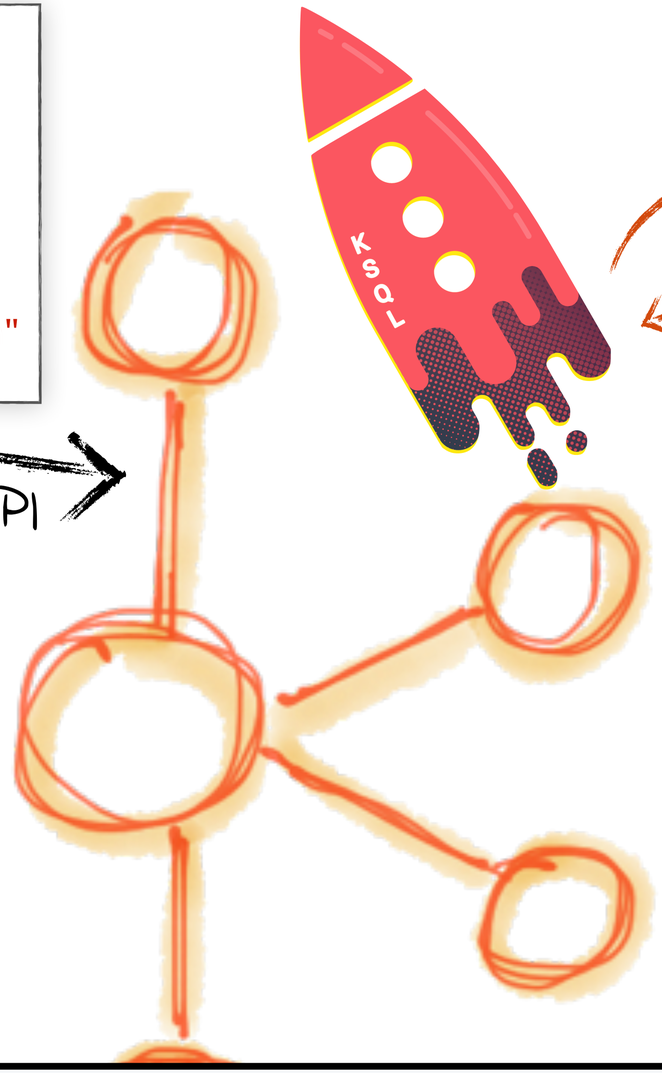



Rating events
App
Producer API
Push notification
App
Consumer API
Kafka
Connect
Operational Dashboard
Elasticsearch
User data
RDBMS
Kafka
Connect
ksqlDB
Join events to users, and filter
Kafka
Connect
Data Lake
SnowflakeDB/
S3/HDFS/etc


Rating events
App
Producer API
Push notification to Slack
App
Consumer API
Kafka Connect
Operational Dashboard
Elasticsearch
Kafka Connect
User data
RDBMS
Kafka Connect
ratings
ksqlDB
poor_ratings
Data Lake
S3/HDFS/ SnowflakeDB etc
Filter events

```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105213,
 "channel": "web",
 "message": "worst. flight. ever. #neveragain"
}
```


Producer API

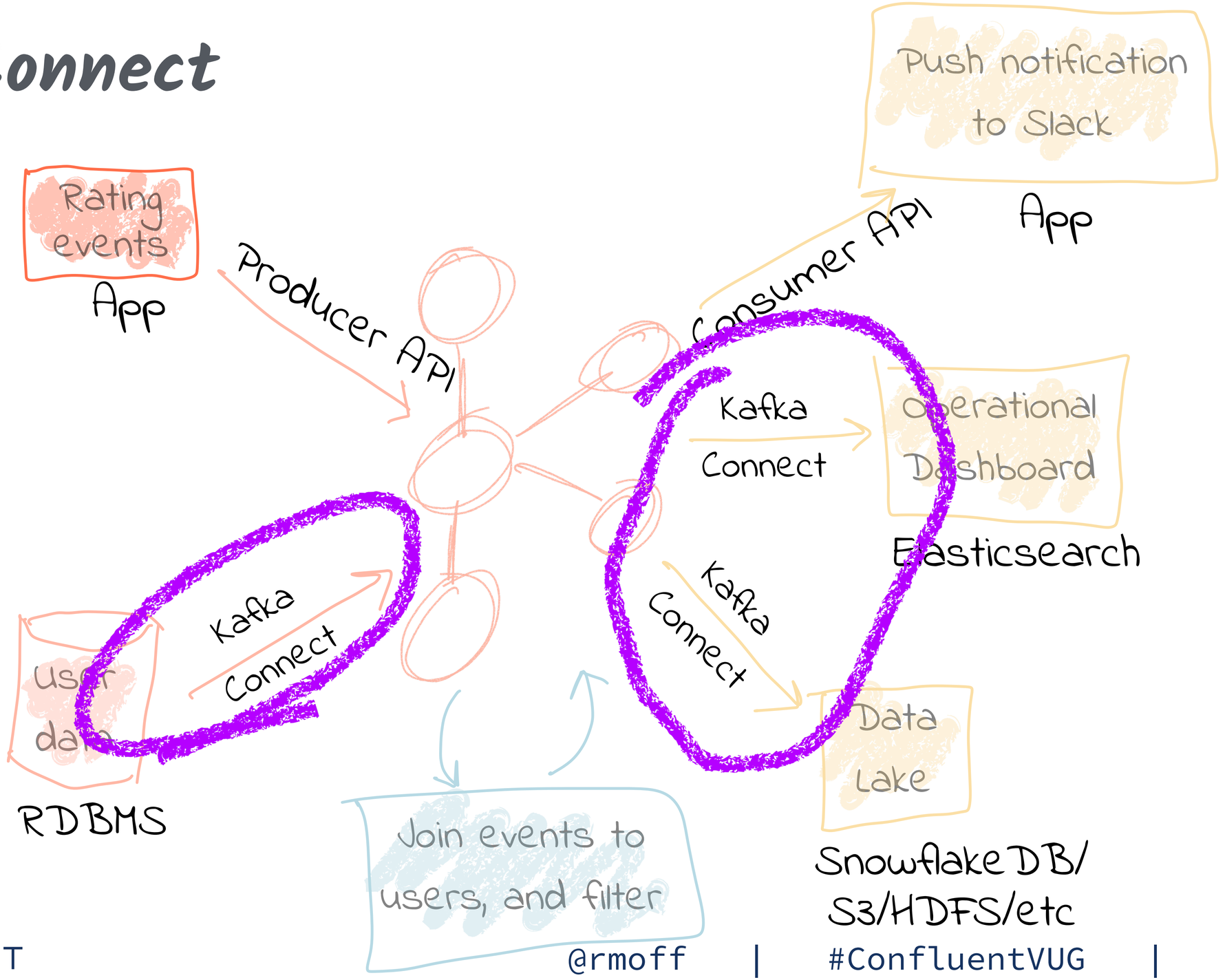


Filter all ratings where STARS<3

POOR_RATINGS

```
CREATE STREAM POOR_RATINGS AS
SELECT * FROM ratings WHERE STARS <3
```

Kafka Connect



```
{
  "rating_id": 2133,
  "user_id": 9,
  "stars": 1,
  "route_id": 7219,
  "rating_time": 1519402815063,
  "channel": "web",
  "message": "worst. flight. ever. #neveragain"
}
```


Producer API

Kafka Connect

Debezium

MySQL


```
{
  "uid": 9,
  "name": "Neha",
  "locale": "en_US",
  "address_city": "Palo Alto",
  "elite": "P"
}
```

The Stream/Table Duality

Stream

Account ID	Amount
12345	+ £50
12345	+ £25
12345	- £60

Time

Table

Account ID	Balance
12345	£50

Account ID	Balance
12345	£75

Account ID	Balance
12345	£15

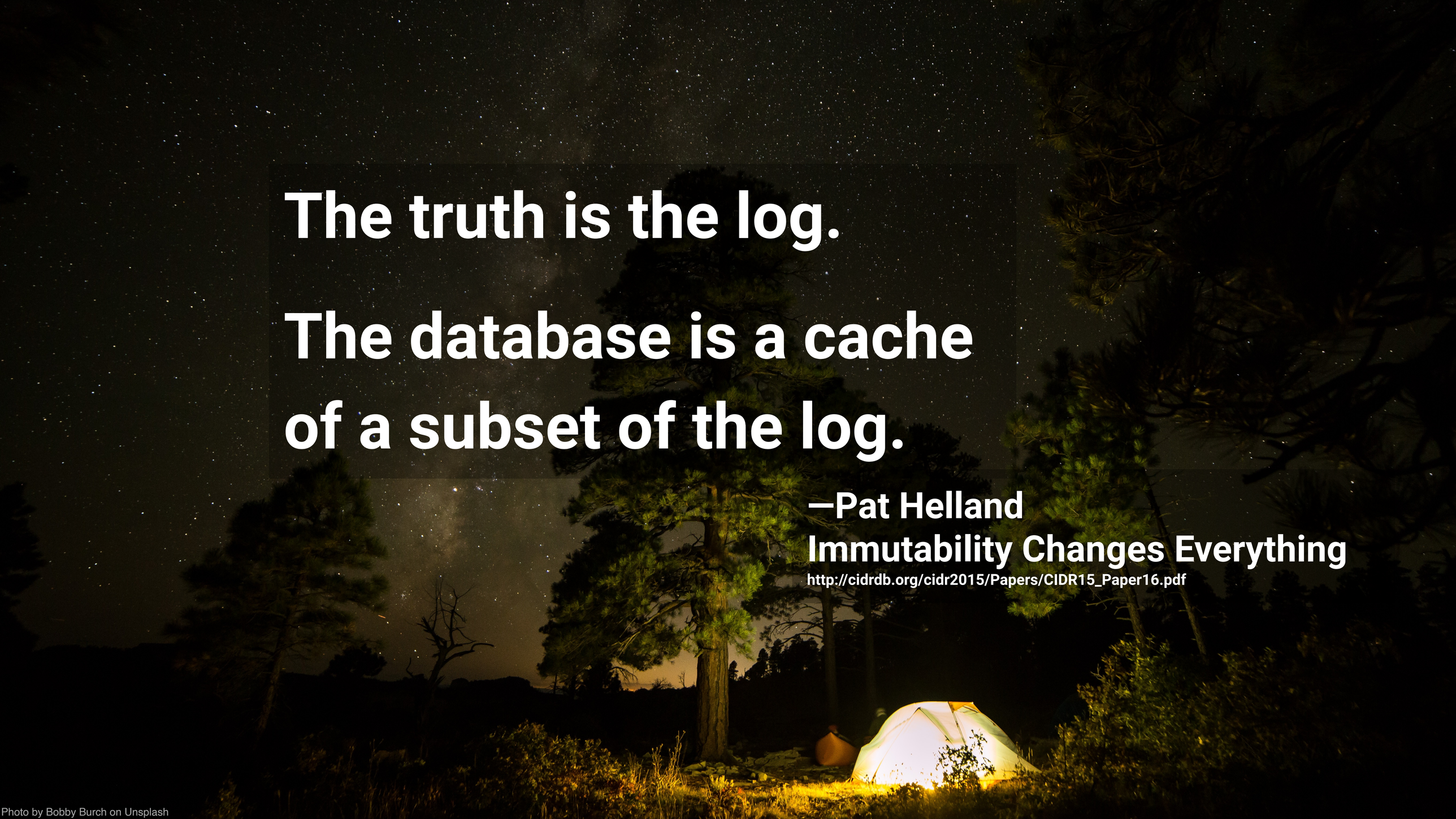
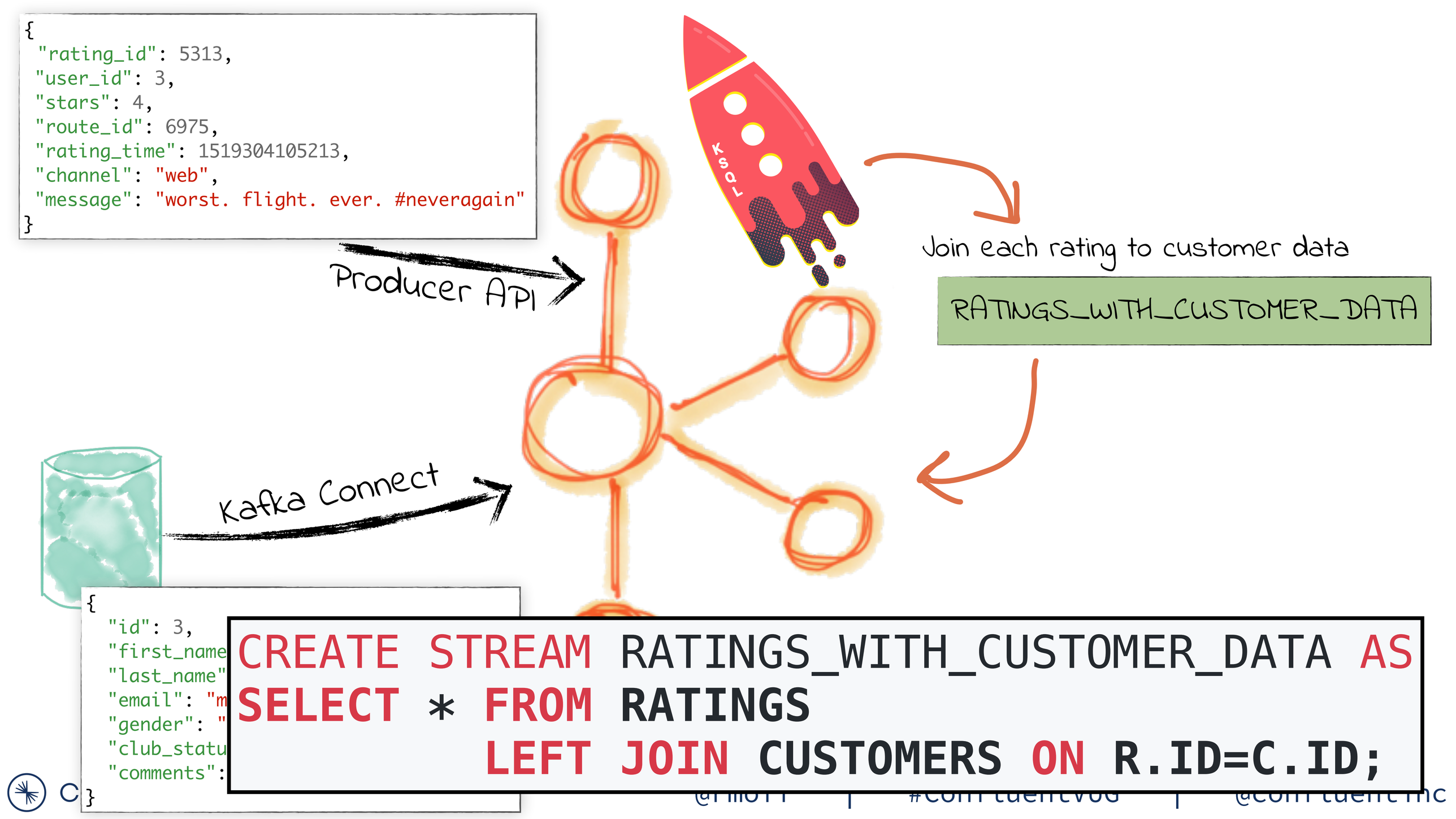
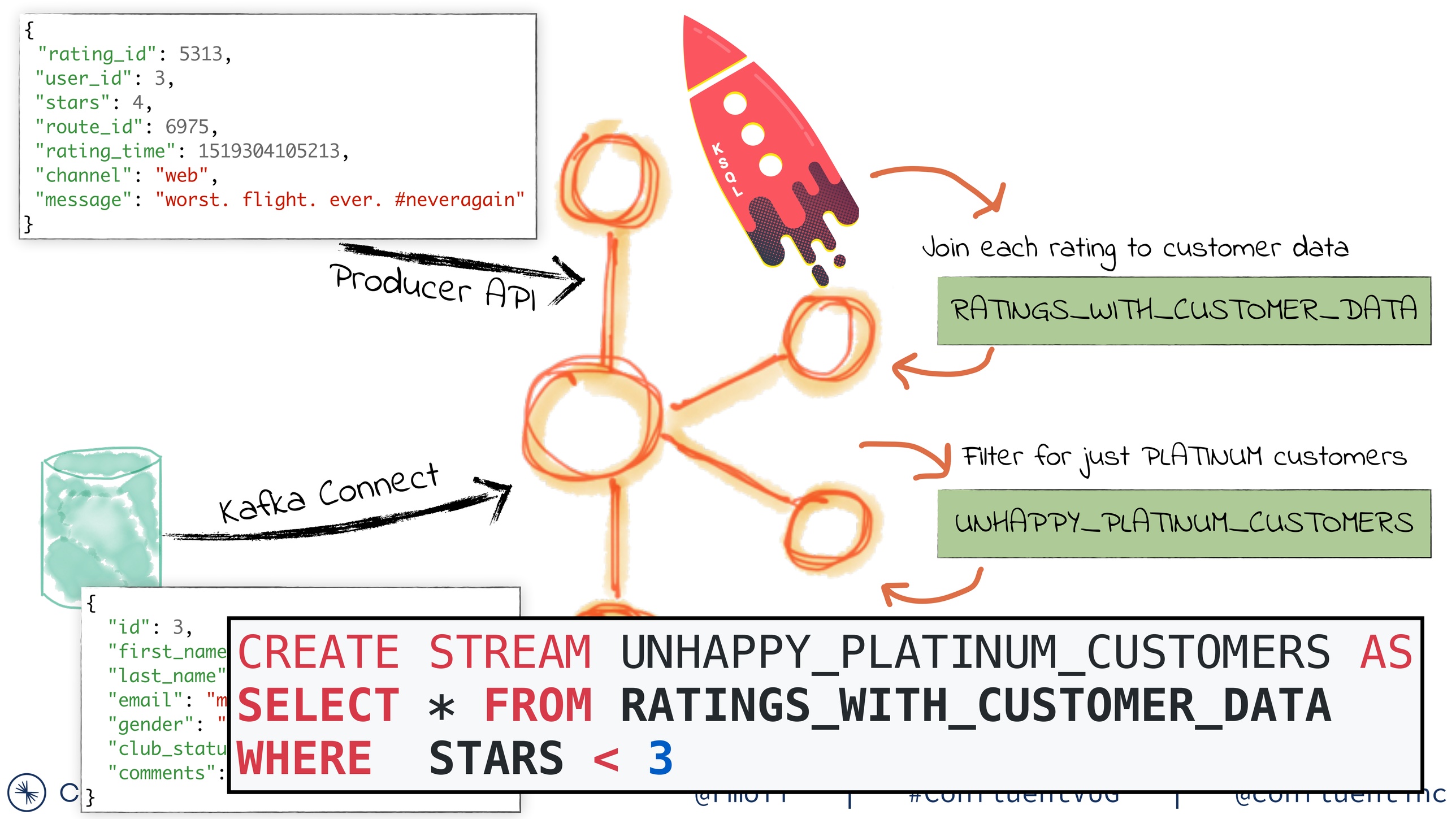


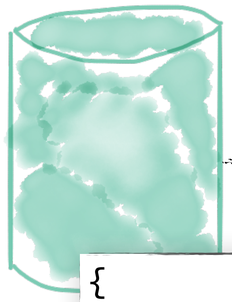
Photo by Bobby Burch on Unsplash


{
"rating_id": 5313,
"user_id": 3,
"stars": 4,
"route_id": 6975,
"rating_time": 1519304105213,
"channel": "web",
"message": "worst. flight. ever. #neveragain"
}
Producer API
KSQL
Join each rating to customer data
RATINGS_WITH_CUSTOMER_DATA
Kafka Connect
{
"id": 3,
"first_name
"last_name"
"email": "m
"gender": "
"club_statu
"comments":
}
CREATE STREAM RATINGS_WITH_CUSTOMER_DATA AS
SELECT * FROM RATINGS
LEFT JOIN CUSTOMERS ON R.ID=C.ID;


{
"rating_id": 5313,
"user_id": 3,
"stars": 4,
"route_id": 6975,
"rating_time": 1519304105213,
"channel": "web",
"message": "worst. flight. ever. #neveragain"
}
Producer API
KSQL
Join each rating to customer data
RATINGS_WITH_CUSTOMER_DATA
Kafka Connect
Filter for just PLATINUM customers
UNHAPPY_PLATINUM_CUSTOMERS
{
"id": 3,
"first_name
"last_name"
"email": "m
"gender": "
"club_statu
"comments":
}
CREATE STREAM UNHAPPY_PLATINUM_CUSTOMERS AS
SELECT * FROM RATINGS_WITH_CUSTOMER_DATA
WHERE STARS < 3

```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304
 "channel": "web",
 "message": "worst. fli
}
```


```
CREATE TABLE RATINGS_BY_CLUB_STATUS AS
SELECT      CLUB_STATUS, COUNT(*)
FROM        RATINGS_WITH_CUSTOMER_DATA
            WINDOW TUMBLING (SIZE 1 MINUTES)
GROUP BY CLUB_STATUS;
```



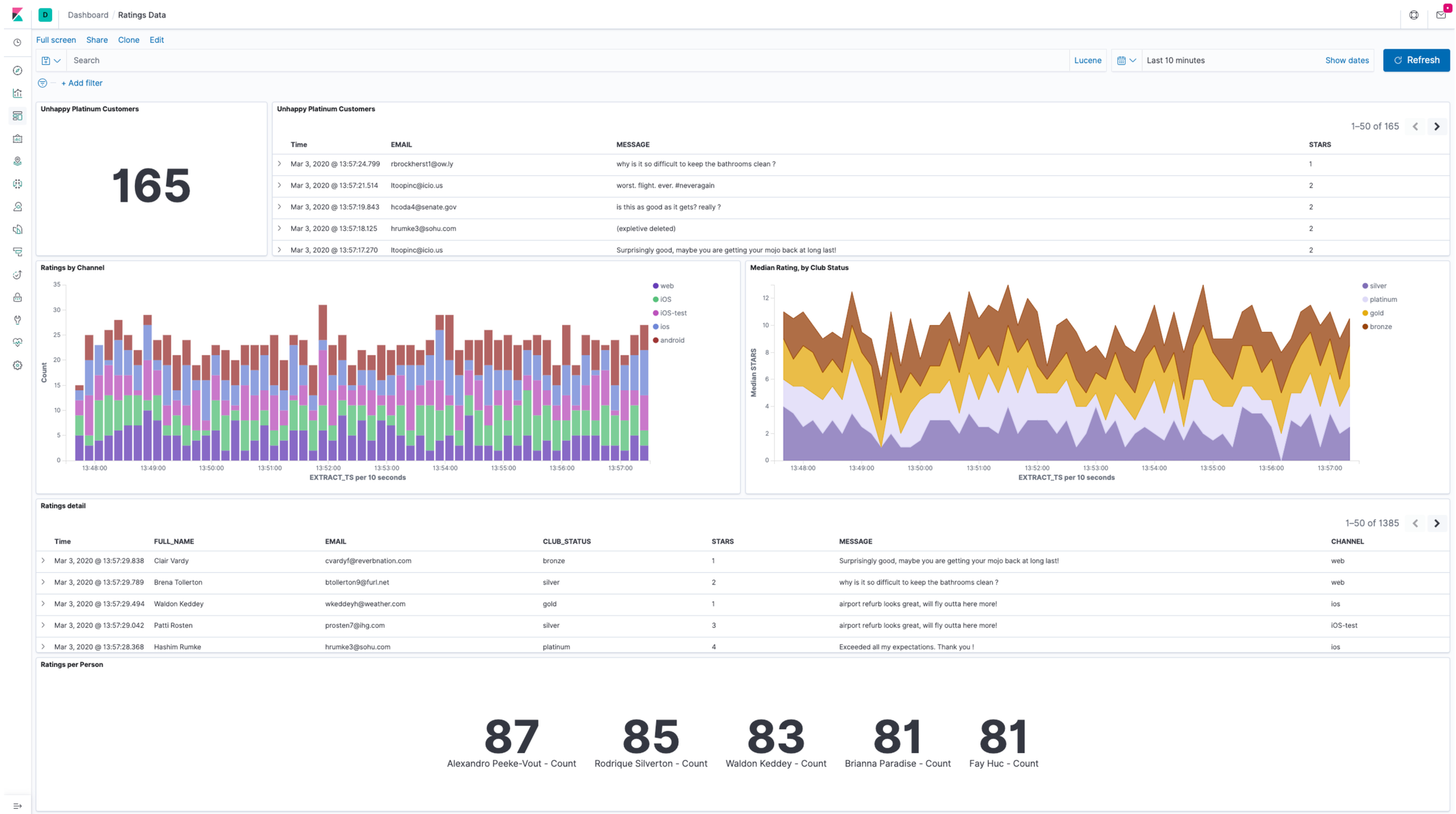
Kafka Connect


```
{
  "id": 3,
  "first_name": "Merilyn",
  "last_name": "Doughartie",
  "email": "mdoughartie1@dedecms.com",
  "gender": "Female",
  "club_status": "platinum",
  "comments": "none"
}
```

Aggregate per-minute by CLUB_STATUS

RATINGS_BY_CLUB_STATUS_1MIN

Kafka Connect → Elasticsearch



DAMOFF

KAFKA SUMMIT 2020

DEVELOPER.CONFLUENT.IO